

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II
SCUOLA DI DOTTORATO IN
INGEGNERIA INFORMATICA ed AUTOMATICA
DIPARTIMENTO DI INFORMATICA E SISTEMISTICA



Grasp and Manipulation of Objects with a Multi-Fingered Hand in Unstructured Environments

Fabio Ruggiero

`fabio.ruggiero@unina.it`

In Partial Fulfillment of the Requirements for the Degree of
PHILOSOPHIAE DOCTOR in
Computer Science and Automation Engineering

November 2010

TUTOR
Dr. Vincenzo Lippiello

COORDINATOR
Prof. Francesco Garofalo

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II
SCUOLA DI DOTTORATO IN
INGEGNERIA INFORMATICA ED AUTOMATICA

Date: **November 2010**

Author: **Fabio Ruggiero**

Title: **Grasp and Manipulation of Objects with a
Multi-Fingered Hand in Unstructured Environments**

Department: **DIPARTIMENTO DI INFORMATICA E
SISTEMISTICA**

Degree: **PHILOSOPHIAE DOCTOR**

Permission is herewith granted to university to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

.... to my parents, to my sister and to Rita

“If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas” - George Bernard Shaw

Acknowledgments

Finally...after three years...here I am: at a conclusion of a fantastic experience for which I have many people to thank.

First, I deeply thank my advisor Dr. Vincenzo Lippiello. Since my master thesis, his patience, kindness and invaluable support have been the key point for my professional growth. He has been my reference point in this whole new world for me and thanks to his passion and devotion for the research I could move on in the most difficult moments.

I am thankful also to Prof. Luigi Villani whose scrupulousness and advices helped me a lot in the development of my researches.

I have to say a huge “*thank you*” also to Prof. Bruno Sicliano: I discovered the “*robotic world*” through his classes. Moreover, he made possible this Ph.D experience and all my travelings all around the world.

A key point in my life has been represented by the period I spent at Northwestern University under the supervision of Prof. Kevin Lynch. In this period I grew up both from a personal than from a professional point of view. For this, I have to deeply thank Prof. Lynch: his patience, kindness, passion and experience helped me a lot beyond the “*simple*” research activities.

Further, I have to mention all LIMS people who made me feel as at home.

I have to thank also my “local colleagues” of the PRISMA group. They enjoyed with me a lot of time also out of the “standard” situations and working days.

I am also grateful to my Doctorate School, in the persons of Prof. Luigi Pietro Cordella first and then Prof. Francesco Garofalo.

I thank also all the professors, young colleagues, students and friends met during these years in summer schools, European projects meetings and in the robotics conferences and seminars.

My family: initially they did not believe this experience could be useful for me, but during these three years they changed their minds. Nevertheless, I’ve never felt missing their support, and this allowed me not to give up.

I am thankful also to all my friends for the nice time spent with them: they distracted me when it was useful to restore and fortify myself. Mention all of them in these few lines is impossible...but they know!

Last but absolutely not least, the biggest “*thank you*” (...if this does not hurt

other people) goes to Rita. Her love, patience and support have been essential for the (hopefully) good results in my research works. Of course, she got angry several times for the week-ends I had to spend studying, but with understanding and love all can be come overcome.

Grazie!

The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

Napoli, November 2010

Fabio Ruggiero

Summary

Year by year, service robotics is becoming an attractive field and a cutting-edge research topic, and it has been able to attract the attention of several international research groups. In such a scenario, the European project Dexmart (FP7-ICT-216239), under which this thesis has been mainly developed, intends to propose new solutions about bi-manual control in the so called “human-friendly” applications.

Operating in unstructured environments for such tasks as grasping and manipulation of unknown objects in a “human-like” fashion, fits quite well in the above concerns: a robot, to be truly a part of service robotics applications about the control of multi-fingered hands, it should be able to perform autonomous works in a dexterous way with or without the presence of humans in the scene.

In this thesis, new methods to grasp and to manipulate unknown objects with a robotic multi-fingered hand have been proposed. These methods have the goals to reduce the time of the execution of the grasp to be suitable with real-time applications and to exploit the redundancy in the “multi-fingered hand + object” system in order to optimize other subtasks during manipulation. In particular, a visual system has been used to cope with the problem of unknown objects grasping. Then, it is straightforward to recognize that two main tasks have to be performed in such a context: object recognition/reconstruction and grasp planning. The motivation behind the proposed method is that the visual reconstruction algorithm should guide the object’s grasp planner in a coordinated manner. The same coordination aspects are the key-points of the proposed method about manipulation of objects: now, the fingers of the robotic hand have to cooperate in order to manipulate the object in the desired way with a good dexterity and this can be done exploiting the redundancy of the whole system.

In detail, the outline of this thesis is organized as follows.

- Chapter 1 is a general introduction underlining the need to make a robot autonomous or at least able to operate in unstructured scenarios to cope with service robotics and industrial applications requests.
- Chapter 2 presents the current state of the art about reconstruction and grasp of objects, as well as grasp planning and manipulation control.

- Chapter 3 introduces a new visual grasp algorithm which is mainly split in two part: object surface reconstruction and grasp planning. As explained before, the chapter is stressed as to point out that the reconstruction surface's evolution guides the grasp of the unknown object in order to reduce the whole grasping time execution. The price to pay is the possibility to reach optimal local solutions with respect to several grasp quality measures. Simulations and experiments are presented to validate the proposed approach.
- Chapter 4 discusses about the use of a closed loop inverse kinematic (CLIK) algorithm in the context of coordinated manipulation tasks. The proposed method adapts the well-known mathematical tools about robotic manipulation to fit into the CLIK framework in order to exploit the properties of the latter about redundancy management for dexterity scopes. Simulations are presented to validate the proposed approach.
- Chapter 5 contains conclusion, remarks and proposals for possible developments.

Contents

Acknowledgements	ix
Summary	xi
List of figures	xvii
List of tables	xix
1 Introduction	1
1.1 DEXMART: a large-scale integrating project	2
2 State of the art	5
2.1 Object reconstruction	6
2.2 Grasp planning	7
2.2.1 Grasp preshaping	7
2.2.2 Grasp quality measures	7
2.2.3 Human-like grasp	8
2.3 Manipulation control	9
2.3.1 Computed torque control	10
2.3.2 Event-driven control	11
2.3.3 Hybrid position/force control	13
2.3.4 Hierarchical control	14
2.3.5 Predictive control	15
2.3.6 Control basis approach	16

2.3.7	Adaptive control	16
2.3.8	Stiffness control	17
2.3.9	Impedance control	19
2.3.10	Kinematic control	21
3	Visual grasp algorithm	23
3.1	Description of the visual grasp algorithm	24
3.2	Preparation stages	26
3.2.1	Acquisition of the images	26
3.2.2	Elaboration of the images	27
3.2.3	Preshape of the object	28
3.3	Reconstruction algorithm	30
3.4	Local grasp planner	32
3.4.1	Initial configuration of the grasp	32
3.4.2	Proposed quality measure for the grasp	33
	Motion field of forces	34
	Kinematic barrier of forces	35
	Evaluation of the quality of the grasp	36
3.5	Generation of the fingers trajectories and motion controller	37
3.6	Simulations and experiments	37
4	CLIK algorithm for dexterous manipulation	41
4.1	Kinematics of objet and fingers	41
4.2	Contact kinematics	42
4.3	Classification of grasps	46
4.4	Kinematic motion control with redundancy resolution	47
4.5	Simulation of the case study	49
5	Conclusion and future research directions	55
5.1	Main results	55
5.2	Proposals for the future	56

A	Appendix	59
A.1	Mirtich and Canny grasp quality index	59
A.2	Re-arrangement of the Mirtich and Canny quality index	60
A.3	Simulations	62
	Bibliography	65

List of Figures

2.1	Types of sensors and their use in manipulation[43].	12
2.2	Manipulation control scheme proposed in [43] with a detector of events and a module for smooth transitions.	13
2.3	Hierarchical control method structure [49].	15
2.4	Adaptive control scheme [40].	17
2.5	Unified control system architecture for multi-fingered manipulation in the framework of stiffness control [37].	19
3.1	Classical serial method vs. proposed parallel approach.	24
3.2	The block diagram of the proposed visual grasp algorithm.	25
3.3	Camera stations (bullets) and trajectories of the camera during the acquisition of the images.	27
3.4	Example of the image processing technique employed to extract the silhouette and the bounding box from an acquired image of the observed object.	28
3.5	Examples of the preshaped elliptical reconstruction surfaces.	29
3.6	Cross network topology of the reconstruction surface with the virtual mass, springs and spatial damper of the i -th sample points.	30
3.7	Concept of the proposed visual grasp.	33
3.8	Force field for the i -th contact point.	34
3.9	Contour of neighbor points of the current target grasp point and possible direction of movements of the associated finger.	36
3.10	Steps of the process about reconstruction of the object's surface: on the left a teddy bear, on the right a bottle.	38
3.11	Trajectories of the fingers generated by the local grasp planner (green: approach, red: grasp) and the corresponding sequence of floating grasp points achieved during the reconstruction process (yellow) for the two objects, both evaluated with $k_{cm} = 1$ (left) and $k_{cm} = 0$ (right).	39
4.1	Local parametrization of the object's surface with respect to the object frame.	43

LIST OF FIGURES

4.2	Block scheme of the CLIK algorithm with redundancy resolution based on equation (4.10). Notice that $N = n_f$	48
4.3	Manipulation system for the case study.	49
4.4	Path imposed to the position of the object.	50
4.5	Time history of the object's pose error in terms of the average of the norm of the pose errors of the 4 parallel CLIK algorithms. Dashed line: without secondary tasks. Continuous line: with secondary tasks.	51
4.6	Time history of joint 3 variable. Dashed line: without secondary tasks. Continuous line: with secondary tasks.	52
4.7	On the left: time history of the distance between tips 1 and 3. Dashed line: without secondary tasks. Continuous line: with secondary tasks. On the right: time history of the x position of the contact points with respect to the center of the object. Dashed line: without secondary tasks. Continuous line: with secondary tasks.	52
A.1	Finger trajectories evaluated by the local grasp planner (continuous lines) and the corresponding sequence of grasp points on the reconstructed object surface (dotted lines).	61
A.2	Finger trajectories evaluated by the local grasp planner (continuous lines) and the corresponding sequence of grasp points on the reconstructed surface for the smooth prism (dotted lines).	62

List of Tables

3.1	Comparison between the numerical evaluations – through the use of some traditional grasp quality measures – of the grasp’s final configurations obtained with the serial approach and the parallel one. For the latter, the final configuration of the grasping points has been found using the proposed quality index.	40
-----	---	----

Chapter 1

Introduction

Over the course of centuries, human beings tried to design and to build new machines first to help themselves in the execution of several tasks, and then to completely replace themselves, especially in the most dangerous works. In a short time, this desire about having machines in substitution of human being in physical activities has been caught up as well by the desire to substitute him in decision making tasks.

The term *robot* derives from the term *robota* which means executive labour in Slav languages. As well, *robotics* is commonly defined as the science studying the *intelligent connection between perception and action* [89]. These two last definitions show how perfectly a robot fits into the above human being's desires: these last, besides, can be accomplished if and only if the *three fundamental laws* introduced by Asimov are respected. These laws established rules of behavior to consider as specifications for the design of a robot, and they are namely:

- A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- A robot must obey the orders given by human beings, except when such orders would conflict with the first law.
- A robot must protect its own existence, as long as such protection does not conflict with the first and the second law.

Nowadays, robots are widely used in industrial applications for such works where human being would have more risk for his life, more cost per hour and more stress for his body. The connotation of a robot for industrial applications is that of operating in a *structured environment* whose geometrical characteristics are mostly known a priori.

Hence, operating in scarcely structured or *unstructured environments* – where the geometrical characteristics are not known a priori –, even with or in cooperation

with humans, it is not possible without robots with a marked characteristics of autonomy. The expression *advanced robotics* usually refers to this framework, in which the ability in decision making tasks plays a relevant role. Advanced robotics is still a young discipline and therein several researchers are motivated to investigate solutions which could be the answer to the growing need of autonomous robots for domestic and service applications, but also for new industrial requests.

Therefore, robotic systems of the next decade will be, potentially, a part of everyday life as helpers and eldercare companions, assisting surgeons in medical operations, intervening in hazardous or life-critical environments for search and rescue operations and so on. Personal and service robots will thus be found in all domains of our future life, and they represent not only a hope for a more convenient world but also a massive new market for leading-edge technology industry and significant business opportunities, especially for industries. Only a few of the technologies required to build functional personal and service robots already exist at the component level and markets for these products are getting gradually into place. Continuous research and development efforts are required to combine the different technologies, create new products and services, enhance the existing ones for a wide range of possible applications.

The realization of a truly dexterous and autonomous manipulation system is still an open research issue: grasp and manipulation of objects with a multi-fingered robotic hand are such complex tasks combining different strategies constraints, goals, advanced sensing and actuating technologies, requiring new concepts and design of artificial cognitive systems.

1.1 DEXMART: a large-scale integrating project

The DEXMART (DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills: A bridge from natural to artificial cognition) European project is founded by the European Commission under the FP7 framework programme. DEXMART has the ambitious mission to fill the gap between the use of robots in industrial environments and the use of future robots in everyday human and unstructured scenarios.

The key innovations which will be carried out within the project are:

- development of original approaches to interpretation, learning and modeling, from the observation of human manipulation at different level of abstraction;
- development of original approaches to task planning, coordination and execution in order to confer to the robotic system self-adapting capabilities and reactivity to changing environment and unexpected situations;
- design of effective control strategies for a dual-hand/arm robot manipulator that can be easily parameterized in order to preserve smoothness during the

transitions at the contact with objects;

- design and development of new actuators, as well as new mechanical structures and materials, able to overcome the limitations of current manipulation devices;
- development of meaningful benchmarks for dual-hand manipulation.

Hence, it is clear from the Summary that this work thesis will be especially focused on the just mentioned second aspect.

The achievement of the research objectives within DEXMART will have an important impact toward the achievements of robust and versatile behavior of artificial systems in open-ended environments providing intelligent response in unforeseen situation.

Chapter 2

State of the art

Grasping and manipulation tasks, in general, require a priori knowledge about the geometric characteristics of the objects and of the scene. One of the first approaches about grasping in unknown environments can be found in [111], where a visual control of the grasp is performed employing visual information in order to track both the positions of the object and of the fingers.

A method to grasp an unknown object using information provided by an algorithm with deformable contour model is proposed in [74]. In [110], an omnidirectional camera is used to recognize the shape of the unknown object, while grasping is achieved on the basis of a grasp quality measure, using a soft-fingered hand.

The previous mentioned works are concerning about the general problem of the grasp in unstructured environments, and in [10] it is clearly stated that the task of autonomous grasping can be generally divided into:

- detection of the object;
- recognition of the object;
- coarse end-effector alignment;
- preshaping;
- vision-guided grasping;
- execution of the desired grasp action.

Starting from this concept, it is straightforward to recognize that the previous actions can be grouped into two main tasks which are namely:

- object's recognition/detection;
- grasp planning.

In this chapter, a survey about these last tasks will be performed. Moreover, since a manipulation action has to be carried out in addition to the grasp of the unknown object, it is necessary to investigate also the literature about control of manipulation tasks, with particular attention to the problems about kinematic redundancy of the “robotic multi-fingered hand + object” systems.

2.1 Object reconstruction

For many years, reconstruction of 3D models of objects has been used to solve a certain number of applications in robotics, reverse engineering, measure and quality control, medical field.

Several methods have been proposed in the literature to achieve this goal, generally based on different approaches for data acquisition, such as omnidirectional cameras, single or multiple cameras mounted on robots, photos taken by hands or with tripod, etc. Some relevant differences are in how the obtained images are processed and, in particular, in the algorithm used to reconstruct the model of the object.

A certain number of algorithms can be classified under the so called *volumetric scene reconstruction* approach [30]. This class of algorithms can be further divided into two main groups, namely:

- *Shape from silhouettes.*
- *Shape from photo-consistency.*

In the former, the silhouette images play a crucial role: binary images are used where the value of a pixel indicates whether or not the visual ray from the optical center intersects the object’s surface. Hence, the union of all visual rays defines a solid cone and the intersection of all these cones for all the images defines a volume in the scene corresponding to a first approximation of the object’s volume. Starting from this last, different techniques to reconstruct the model of the object have been developed. One of the most used methods samples the volume in voxels [30]: in this case, the task consists in creating the description of the voxel’s occupancy corresponding to the intersection of the back-projected silhouette cones. To have a more efficient description of the scene, the voxels are organized in an octree representation: how to build this octree is an important research topic [42, 71, 97].

In the latter, gray-scaled images or colored ones are used. These further information are employed to improve the 3D model reconstruction of the target object: *voxel coloring* [87] and *space carving* [54, 105] are some among the most investigated techniques in this field.

Several methods, instead, can be classified under the so called *object’s surface reconstruction* approach, since only the surface and not the volume of the object is needed to accomplish the reconstruction. In [104], the model of the object is

obtained starting from a surface that moves towards the object under the influence of internal forces, given by the surface itself, and external forces, given by the image data. Typically, the starting surface is a sphere. This approach may be considered as a generalization of snakes used in 2D.

The finite-elements method is used in [19] to reconstruct both 2D and 3D boundaries of the object. Using an active contour model, data extracted from taken images are employed to generate a pressure force on the active contour which inflates or deflates the curve, making its behavior like a balloon.

A technique for computing a polyhedral representation of the *visual hull* – the set of points in the space that are projected inside each image silhouette – is studied in [35]. In this approach, only the contours of the silhouettes in the images have to be visited, and the computed visual hull is quickly represented.

Furthermore, other methods rely on the use of *apparent contours* [18, 80]: in these cases the reconstruction is based on a spatio-temporal analysis of deformable silhouettes.

2.2 Grasp planning

In this section, a survey about grasp planning techniques has been performed. First, the methods which are present in the literature about grasp preshaping are briefly described, then algorithms used to select suitable grasp points on the objects are presented and, in the end, several techniques concerning the choice of a human-like grasp are illustrated.

2.2.1 Grasp preshaping

Preshaping of a robotic multi-fingered hand – the preparation of the hand to the grasp of the object – is a non-trivial step before the grasp planning, as described in [70].

In the literature, several methods deal with the preshaping problem, and most of them rely on a previous knowledge learned from humans [96]. Others rely on the use of vision [103], adopting a variety of approaches such as fuzzy logic [4] or box-based approximation [47], and they are generally task-dependent [81].

2.2.2 Grasp quality measures

Grasp planning techniques rely upon the choice of some grasp quality measures used to select suitable grasp points. Several quality measures proposed in the literature depend on the position of the contact points (algebraic properties of the grasp matrix, geometry of the grasp area of the polygon created by the contact points and so on), while others depend on the finger forces or on the external resistant wrench.

In [32], two general optimal criteria are introduced, where the total finger force and the maximum finger force are considered, while in [64] simple geometric conditions to reach an optimal force closure grasp both in 2D and in 3D are found. The geometric properties of the grasp are also used in [55] to define some quality measures, as well as suitable task ellipsoids in the wrench space of the object have been proposed to evaluate the quality of the grasp also with respect to the particular manipulation task to accomplish.

A geometrical approach obtaining at least one force closure grasp for 3D discretized objects is studied in [84], where two algorithms are investigated: the first finds at least one force closure grasp, while the second optimizes it to get a locally optimum grasp.

Measures depending on the configuration of the hand [88] define a set of quality measures based on the evaluation of the capability of the hand to realize the optimal grasp. A rich survey of these grasp quality measures can be found in [95].

In order to plan a grasp for a particular robotic multi-fingered hand, quality measures depending both on the geometry of the grasp and on the configuration of the hand should be taken into account. Few papers address the problem of grasping an unknown object using a given robotic hand, able to reach the desired contact points in a dexterous configuration [11, 15, 33, 36, 39].

A grasp control task is considered in [75], where several controllers are combined to reach different wrench closure configurations, while in [78] grasp prototypes – generalization of example grasps – are used as starting points in a search for good grasps.

2.2.3 Human-like grasp

In general, human beings can grasp and manipulate a large variety of objects with a high level of dexterity. An elaborated taxonomy of human grasps can be found in [22].

Throughout the literature, two main approaches among the others can be recognized in the process of transferring human manipulation skills to robotic multi-fingered hands. Namely, they are:

- *Programming by demonstration.*
- *Neural networks and genetic algorithms.*

Considering the former technique, movements of human beings are recorded and analyzed off-line using a motion capture system, and therefore the motion is transferred to a robotic hand [73]. Further, it has been demonstrated that humans perform different grasps in reason of the task's specifications, even if the orientation and the location of the objects are kept the same. A programming by demonstration system, which shows how fine manipulation tasks (e.g. screw

moves) can be recognized, it is presented in [112], where the recorded trajectory is analyzed, interpreted and mapped to a manipulator.

Considering the latter approach, the space of all feasible grasp configurations is analyzed using genetic algorithms [13]. Since these last are not suitable for real-time applications, neural networks have been adopted. In this way, a neuro-genetic architecture is employed in the sense that the genetic algorithms are used to create a training set for the neural network. In [34], a human-like grasp is recognized by a biologically plausible neural network. This last is built upon a hierarchical model for motion detection using a view-based recognition approach which is consistent with principles in the human cortex.

Further approaches for detecting and performing human-like grasps are presented in [1, 2], where a *qualitative reasoning* approach to the synthesis of dexterous grasps is provided. An intelligent planner has been developed in order to perform this synthesis, advantageously adopting qualitative methods instead of analytical or numerical models. However, only coarse solutions can be provided, since this approach is an attempt to strike a compromise in the use of qualitative and quantitative resources.

When human-like grasps must be achieved in unstructured environments, real-time performances are necessary and no pre-recorded trajectories are available. Hence, some visual sensor has to be considered to reconstruct the object to be grasped and, potentially, manipulated. In [45], kinematic parameters of the human grasp, such as path and preshape, are determined by the three dimensional geometric structure of the target object, and not by the two dimensional projected image of the same object. Moreover, human object recognition is based on identifying coarse structures rather than specific features, as underlined in [9].

2.3 Manipulation control

After the grasp of an object, a typical manipulation task requires the transfer of the same object from a starting position to a goal configuration, avoiding collisions with obstacles, not exceeding the limits of actuators and joints, and supplying proper grasping forces in order to ensure the stability of the grasp during the manipulation.

Since a manipulation task can be subdivided in many events (i.e., change of contacts status, change of contacts type), different control laws may be required for each of them. The design of the feedback control should be therefore integrated with the design of the task planner or widely interact with it.

Basically, the control law problem is the determination of the required joint forces and torques in order to achieve the desired planned manipulation task. Moreover, the controller needs to be robust to deal with uncertainties arising from system modeling, actuators inaccuracy, not modeled events, unknown parameters

and so on. In addition, both the planner and the controller need to be robust with respect to the noise of the sensors.

Starting from these considerations, in this survey, a collection of papers dealing with the control of a single robotic hand has been considered, without any discussion about problems arising with the mechanical construction of robotic hands or with the kinematics and the constraints in the “hand + object” system. In [8, 67], an interesting starting point to cope with these issues can be found.

Assuming this background, starting from the early '90s, many approaches for the control of robotic hands have been reviewed. The major control approaches noticed throughout the referenced papers can be subdivided in these main classes, which are namely:

- Computed torque control.
- Event-driven control.
- Hybrid position/force control.
- Hierarchical control.
- Predictive control
- Control basis approach.
- Adaptive control.
- Stiffness control.
- Impedance control.
- Kinematic control.

In the following, a brief description of these classes is reported.

2.3.1 Computed torque control

The computed torque approach relies on the precise knowledge of the model of the system, which is composed by the robotic hand plus the object and the related constraints. Canceling the whole nonlinear dynamics of the system, it is possible to asymptotically track a desired trajectory for the object's center of mass or, with the assumption of rigid object, for any other point of the object. In the practice, the knowledge of the model, friction parameters, location of the center of mass, etc., is not available and this technique can be useful in simulations. Hence, the need of a precise model for the whole system often involves the use of some other devices to manage robustness problems – like adaptive control schemes.

In [21], a computed torque control law is used to track a desired trajectory for the object's center of mass and to maintain the contacts with no slipping, producing nonzero contact forces which lie within the friction cones: rolling contacts can also be included in this framework. The proposed control law presents three main components: delation of nonlinear terms, introduction of proportional and derivative feedback terms in order to have a decoupled system, and a term which is projected in the null space of the grasp matrix with the effect of regulating the internal forces in order to avoid the slippage.

A computed torque control law with force feedback for compensating uncertainties of the system is presented in [17]. The commanded joint torques consist of two parts: one to move the fingers and the other one to grasp the object. The desired internal forces can be found by optimizing the friction angles, in order to avoid slipping, so that fingers can grasp and manipulate the object in a stable fashion along a desired trajectory. Moreover, a history-based method based on the force feedback is proposed to compensate the model of the system.

With the assumptions that the fingers are not in a singular position, the contacts between the fingers and the object are rigid point contacts, and that the geometry of each body is known, a computed torque law is derived in [14], including kinematics of both rolling and sliding contacts. The commanded joint torques compensate the whole dynamics of the system and, furthermore, a PID controller is employed for the asymptotic tracking of the object's pose, while a PI controller is used for the asymptotic tracking of the desired internal forces.

2.3.2 Event-driven control

As pointed out in [44], much of human dexterity and adaptability to the changes in the task or in the environment is due to the ability of using tactile information to control the process: hence, human manipulation is event-driven. In the same just mentioned work [44], the difficulty of a manipulation control with a hand equipped with tactile sensors is studied. These sensors can be useful to acquire information about contact and object's surface. When contact conditions or task requests change, the control law has to change too and, further, the controller should have smooth transitions between each change of phase. In the mentioned work, different control laws (force control, position control, stiffness control) have been implemented in orthogonal movements directions, and the switching between these controls has been made according to the change of state of the tactile sensors.

Hence, it is clear that the main role in event-driven control is played by sensors. In [43], two figures (Figure 2.1 and Figure 2.2) are drawn to show, respectively, the use of the touch sensing in manipulation and the control architecture for the integration of this tactile sensing.

In the event-driven approach, the detection of events is obviously a crucial step and, in theory, for each of them a proper sensor should be used. In [23]

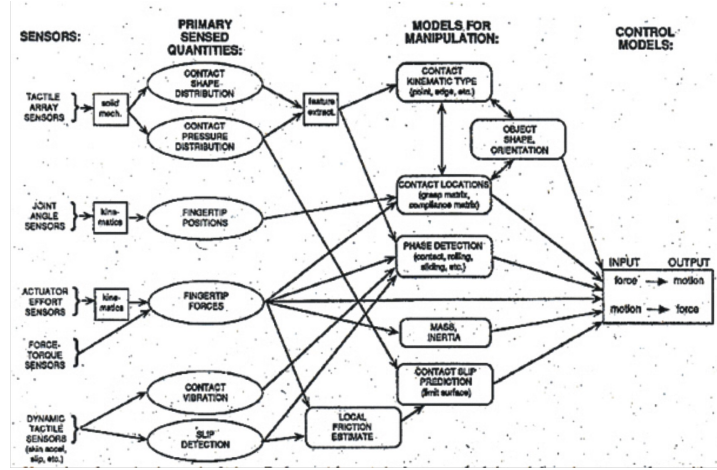


Figure 2.1: Types of sensors and their use in manipulation[43].

there is an useful table showing common manipulation events and sensors which can detect them. Moreover, in the same work, a language based on discrete event systems is introduced in order to model the concept presented in [44] of phase/event/transition. In this language, the time is important to understand when a detected event is the cause for a transition to one phase to another phase. These phase transitions, as said before, correspond to a transition between different control laws: in [23, 51] control schemes to cope with these problems are presented.

Furthermore, in [48], the events considered in the previous phase / event / transition model and the related language can now be also unexpected. The challenge for the authors is to use quicker sensors to acquire more data in the same time of before, in order to have better reactions to the change of events.

A practical application about the event-driven control can be found in [99], where a position control law is performed before that an impact occurs, and in that case the system becomes controlled in force – the force recorded by a physiotherapist is given as set point to this type of controller. The former control law is implemented through a PID controller, while the latter is implemented by a linearization of the dynamics of the system and by a PI controller in order to track the desired values.

From this brief survey, it is clear that event-driven control approach is the link between an higher level controller – optimal grasp selection, grasp planner, etc. – and a lower level controller where there is the practical implementation of the control law. In the middle level, sensors play a crucial role in determining the implementation of the lower level controller, on the basis of data acquisition and

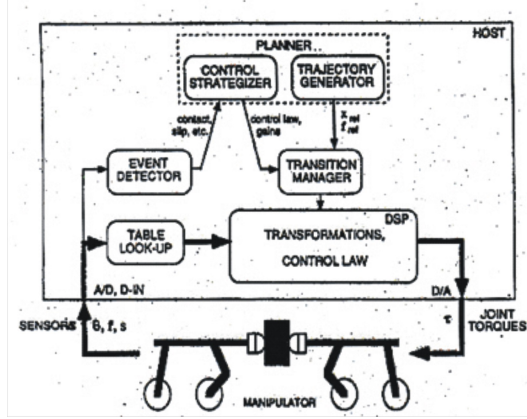


Figure 2.2: Manipulation control scheme proposed in [43] with a detector of events and a module for smooth transitions.

information coming from the upper level.

2.3.3 Hybrid position/force control

Grasping and manipulation tasks require both a force and a position controller to accomplish the desired object's motion, avoiding slippage and satisfying friction constraints. This type of controller can be feasible only if an appropriate decoupling between position and force spaces is performed.

In [92], the position is the main variable to control, while the force is supplied by parallel controllers. The position control algorithm is outlined for both joint space and Cartesian space applications. The robustness of the position controller is guaranteed, while the parallel force control is based upon the requirements for the particular task. Moreover, in the work just mentioned, a final discussion about errors due to the static friction of tendons is presented.

A simple method for a dynamic manipulation/grasping controller of multi-fingered robot hands is proposed in [68]. Starting from the consideration that the force on a finger can be seen as the sum of two orthogonal components, namely manipulation forces – necessary for the manipulation/grasp of the object – and internal forces – necessary to satisfy the constraints about friction cones –, an on-line linearization of the dynamics of the whole system is introduced in order to design a simple controller for manipulation purposes. A PID plus feedforward is used for the position control part, while a PI is employed for the control of the force.

For the rejection of noises in force sensors data, a fuzzy controller has been used in [27]. In particular, the output of the fuzzy controller can perform a correc-

tion on the normal component of the measured force, on the basis of some fuzzy interferences of the measured component of the tangential force.

Another method based on the linearization of the system is proposed in [25]. The dynamics of the system is derived in joint space so as to avoid the kinematic inversion and have a set of differential equations with algebraic constraints. Using some tools present in [50], a state-space input/output linearization is achieved in order to decouple position and force control problems. Moreover, via a pole-placement method, it is possible to give to the system the desired behavior. Further, it has been shown that the controller exists in all the space, except in the singularity points of the fingers. This approach requires the use of force sensors.

The approaches proposed in [28, 31] can be also enumerated in this class of controllers. In those works, the problem of stable grasping and manipulation is faced using a finger pairs covered with a soft compressible layer material. The controller output is a linear combination of signals, each of one addressing a particular subtask, namely: internal forces, external torques balance, desired position of the object's center of mass, etc. Moreover, a high level controller is assumed to manage some grasp quality measures.

Finally, starting from the consideration that if the control of the hand is centralized and the object changes, the controller should change consequently: in [83], a hierarchical controller is implemented, and it achieves a decentralization in order to control each finger in an independent way through a hybrid position/force control scheme.

2.3.4 Hierarchical control

A hierarchical control method can be, in general, included in the previous sections about hybrid position/force control and computed torque, since one of the objectives of a hierarchical controller is to decouple force and position spaces. But some works about hierarchical control have something more: in particular, they take inspiration from human motion control.

In [41], it is pointed out the question of how human brain could control a system like the human body with so many different degrees of freedom interacting in such a complex fashion. Such a complexity is also present in the robotic hands.

Starting from these studies, in [67], a hierarchical control scheme for robot manipulation is built, reflecting a possible hierarchical control scheme for a human finger. The robotic hand is modeled as a set of entities called *robots*, which are coupled to each other and to an object through a set of constraints. The controller has to modify the properties of these entities with the desired attributes. Attaching all these modules, a hierarchical control structure grows up.

Instead, the work in [49] is inspired by two classes of human motions, namely reflexes and voluntary movements: these last suppress a reflex when needed. In this way, a control scheme made up of three modules has been implemented: the

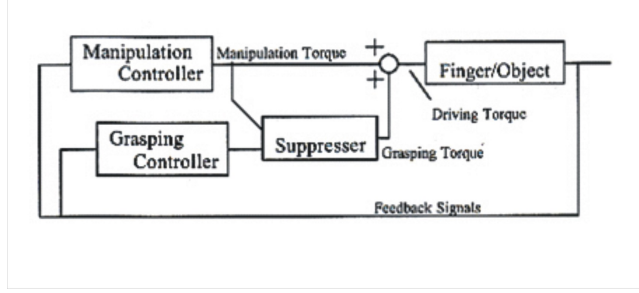


Figure 2.3: Hierarchical control method structure [49].

first one is the grasp controller, which generates torques to avoid the slippage; the second module is the manipulation controller, which generates torques to manipulate the object according to the desired trajectory of the object; finally, the third one is the suppresser module which has the duty to suppress the grasp controller when the manipulation torques become high. Hence, the suppresser is a sort of supervisor managing possible conflicts between manipulation and grasp torques. Figure 2.3 shows the control scheme corresponding to this method.

2.3.5 Predictive control

There is a difference between the hybrid position/force control law and an hybrid model of a robotic hand: in the former case, the model is written in the continuous state space and the control law is said to be hybrid since it is the sum of two different spaces, namely the position space and the force space; in the latter case, the system “hand plus object” is modeled with an hybrid combination of continuous and discrete or logical variables.

This last case is widely studied in the works [106, 107, 108, 109]. In these, the whole dynamics of the system “hand plus object” is recast in a framework subject to complementary and slackness conditions to create a DC (dynamic complementary) model of the system, which includes both continuous and logical variables. This DC model can be represented in a mixed logical dynamical form (MLD model) which allows a unified approach for mathematical, numerical and control investigations. In a MLD model, it is possible to specify actuators limits, joint ranges, and to model the impact of a finger with the surface of the object. The main advantage of MLD models is the possibility to use model predictive control (MPC) schemes that are based on the receding horizon control approach. Moreover, mixed integer quadric problem (MIPQ) solvers can be useful from the numerical analysis point of view, in order to reach an optimal solution for the manipulation control problem. The optimal sequence of both continuous and logical variables is simultaneously found and it corresponds to the optimal motion as to realize dexterous

manipulations.

The theoretical background of MLD models and MIPQ solvers can be found in [5].

2.3.6 Control basis approach

Inspired by linear algebra theory where any vector of the space can be expressed with a linear sum of vectors of the basis, the control basis approach allows to build any controller simply combining controllers of the basis. In such combination, each controller is in the null space of the others guaranteeing the robustness.

A real problem is how to select the simple controllers of the basis, since from this last depends the variety of control that one can apply to the manipulation task.

In [46], a finite states scheme is presented for manipulation control and re-grasp tasks. A combination of controls from the basis corresponds to each state and the basis is made up by three simple controllers, namely: a space motion controller, for the control of the motion in the operational space in order to avoid obstacles; a contact configuration controller, managing the contact forces; a posture controller, for the kinematics of the system in order to manage its redundancy. Each controller of the basis is simple, inner closed-loop by feedback and stable. A combination of simple controllers is also stable. A formal language to deal with this class of control has also been developed. The work in [77] is inspired by this way to control some manipulation tasks with robotic hands, too.

2.3.7 Adaptive control

Adaptive control is often employed when the parameters of the model of the system (hand plus object) are not known at all or they are partially known, and an estimation process is then necessary. Hence, with this technique, the control schemes can deal with unstructured environments and become more flexible for real and practical situations.

In [40], the problem of how to manipulate an object without knowledge about its dynamic and inertia parameters is considered. The proposed scheme is made up of two steps, namely: in the first step the object's center of mass is estimated with an adaptive nonlinear scheme, while in the second step the estimated parameters are used to optimize finger velocities and internal forces, in order not to allow the slippage of the object and to follow desired motions. The adaptive control scheme used in [40] is depicted in Figure 2.4, and the same was used in [6, 90], while the controller used in the latter step is a PD with an acceleration feedforward with a forces optimization module.

Considering sliding contacts and unknown parameters of the object and friction coefficients, in [100], it is proposed a coordinated control scheme for multi-fingered

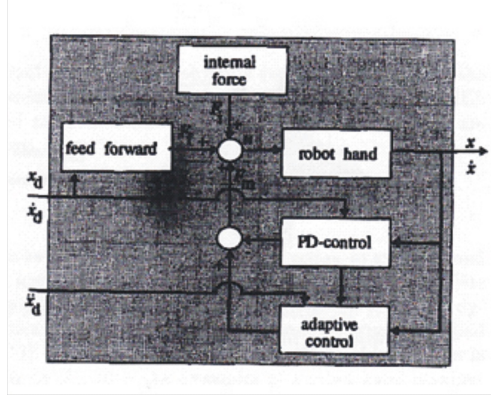


Figure 2.4: Adaptive control scheme [40].

robotic hands. With the assumption of sharp contacts and the hypothesis of objects and fingers described by a two-times continuously differentiable hypersurfaces, the goal is to supply proper torques to the joints as to achieve desired values for the set points. A force sensor is needed in order to measure the contact forces.

A control scheme which is in halfway between a hybrid position/force controller and an adaptive one is proposed in [98]. In this last mentioned paper, two control laws are developed. In the first one, the controller switches quickly and smoothly between a position and a force control scheme on the basis of the current value of the contact force. Moreover, object parameters and disturbance torques are estimated by an adaptive control scheme and then they are used in a proper manner in the control law. In the second control law, tactile sensors have been used in order to measure contact forces so as to avoid slippage. Once again, this information is used in the adaptive control scheme estimating the unknown parameters.

2.3.8 Stiffness control

Compliance or stiffness are among the most important quantities for characterizing a grasp with robotic hands. This is particularly true in fine manipulation, where small motions and low velocities lead to dynamic equations that are dominated by compliance (or stiffness), friction and contact conditions [24].

The stiffness of a robotic grasp represents the rate of change of the grasp forces with respect to (small) object motions. In [24], the dependence of the grasp stiffness towards geometry, contact conditions and mechanical fingers proprieties has been shown. In general, the stiffness of a grasp is a linearization of the expression linking grasp forces and the resulting object's movements. The stiffness matrix includes some effects such as actuators compliance, structural compliance, changes in

grasp geometry, coupling between joints or fingers, different contact types, changes of contact locations and so on.

The work in [24] is aimed to control the gains of actuators in order to have the desired compliance or the desired stiffness matrix. Authors noticed that the stiffness matrix is decoupled into two terms: in the first one it is pointed out the dependence of the matrix with respect to the changes in the grasp's configuration, while the second term expresses the restoring forces at contact points caused by the movements of the object. A method to calculate these sub-matrices is shown: hence, the actual stiffness matrix is the sum of these two components. It is worth noticing that if the eigenvalues of the stiffness matrix are all positive, then the grasp is stable. After calculating the actual stiffness matrix, a sort of error between the actual and the desired matrix is computed and the servo gains are tuned respecting contacts and joints constraints.

A decentralized stiffness control scheme is presented in [16], where the stiffness matrix is evaluated as a sum of two terms like before, and where it is pointed out the dependence also with respect to the changes in the configuration of joints. Rolling and sliding contacts are not included in the formulation, as well as fingers and transmissions couplings. The decentralized object's stiffness control allows the system to reach the desired stiffness in two steps: in the first step the stiffness matrix at fingers level and its reflection at the joint level are calculated with a least square algorithm; in the second step the component in the null space of the previous computed matrix is summed to the previous contribution, and then the gravity forces are compensated.

The stiffness control has obviously some limitations which are pointed out in [91]. Many of these limitations in performance are due to mechanical proprieties of the robotic hands caused by backlash, difficulty in coordination of so many degrees of freedom, tactile and force sensors inadequacy. Other limitations are underlined studying the case of changing the center of compliance of a grasped object. Using the Cartesian stiffness control method, the controller measures the difference between the actual and the desired position of the center of compliance. A restored force is then calculated to carry back the center of compliance in the right position. Through the use of the grasp matrix, this force applied to the object is mapped on a finger force and then in a joint force, through the finger Jacobian. In this simple case, errors can be present in the computation of the kinematics of the hand and of the grasp matrix, due to the presence of rolling and sliding contacts. Moreover, friction constraints put a lower limitation in the feasible stiffness, while the grasp geometry and fingers compliance put an upper limitation.

To overcome some problems pointed out in the previous mentioned work, the same authors proposed in [91] a method in which tactile sensors can be helpful to know the initial pose of the object in order to track the fingers in rolling and sliding movements. The control law is the same proposed in [91], but now it is

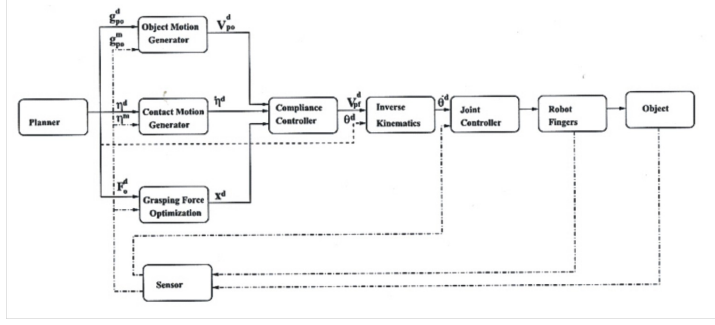


Figure 2.5: Unified control system architecture for multi-fingered manipulation in the framework of stiffness control [37].

assisted by sensors and it is demonstrated that these last reduce errors due to disturbances which act on the stiffness.

The compliance control method is used in a more general framework presented in [37], where a unified architecture (see Figure 2.5) for planning and control is created. The main purpose of the planner is to make the object able to track a desired trajectory, managing the presence of possible obstacles or disturbances, while the controller should be able to be robust with respect to not modeled situations or noise of sensors. The input of the compliance module is the desired motion of the object and the location of the contacts, as well as the desired applied force. With a desired compliance at finger level, all these information are mapped in finger velocities which have to be actuated by a joint control.

The problem of decoupling fingers and joints is faced in [52] with a two-steps solution. In the first step, the relation between stiffness in the operative space and in the fingertip space is pointed out: resolving a linear problem, it is possible to decouple the fingers in order to have a proper controller for each of them. Furthermore, in this step, it is necessary to verify also the conditions on the stiffness matrix: as a matter of fact, the dimensions of the matrix depend on the number of the fingers of the robotic hand and on the dimension of the operational space. In the second step, the relation between stiffness in fingertip space and joint space is found in order to decouple the joints of the same finger. In this way, it is possible to implement a proper compliance controller for each joint.

2.3.9 Impedance control

Impedance control is one of the main approaches used in literature and not only in the field of robotic manipulation. An impedance controller typically allows the robot to act as a mass-spring-damper mechanism with preset parameters.

As pointed out in [86], when object's motion plays a crucial role in the task, an

object impedance approach could be a good solution. In the just mentioned work, grasping and manipulation problems are brought back to a cooperative problem between manipulators. Knowing in advance the dynamics of the whole system, all the “arms” are decoupled and the resultant scheme is linear: hence, an impedance control is implemented in order to achieve a desired impedance at object level, and not at the level of the end-effector of the “manipulators”.

In [69], a robotic hand mounted on a robotic arm is considered. The proposed control law tries to take into account the advantages to have both a robotic arm and a robotic hand. The control scheme makes first a linearization of the system, then it performs an impedance control law in order to give to the system the desired mechanical behavior.

Nevertheless, one of the main advantages in using the impedance control is described in [94], where the proprieties of strictly passivity in steady-state situations of an impedance controller are shown. Moreover, a force control is meaningful only if there is a contact, and then an impedance controller strictly passive can be useful in all situations.

Starting from these considerations, in the last mentioned paper, a virtual object with a virtual center of mass is introduced: the real center of mass is linked both with the virtual one through a spring and with the fingertips through some other 2D or 3D springs. Damper elements are inserted in a proper physical way: in general, they require the measured joints velocities for their implementation, but if these velocities cannot be available, it is hence possible to create a force proportional to the twist acting on the virtual object instead of implementing an observer. In this way, the force is transmitted through the springs and it is also dissipated, creating the desired damping effect.

An impedance controller can also be found in [85], where algorithms for grasp force optimization are studied. A specific module of the control scheme calculates the optimal grasping forces according to various situations that could happen in a manipulation task, such as re-grasping, fingers impact with the object, no motion of the contact points and so on. This module is placed into a control scheme in which each finger is controlled separately by an impedance controller.

To overcome some disadvantages related to fine manipulation – such as joints friction or singularity position –, a new control scheme has been proposed in [7], where first a force sensor output is filtered to avoid noises, then an admittance model computes the desired displacement. Finally, a position control generates the finger joint torques. The admittance model is a mass-spring-damper system, whose matrices are specified through a singular values decomposition.

Lately, the work in [93] about virtual object has been developed in [101, 102]. In these last, the chosen control at object level allows a simpler definition of grasping forces and a compensation of object and robot inertias. Hence, the object is controlled by an impedance controller which is intrinsically passive. The fingers are linked through virtual springs to a virtual object which is in turn linked to

a virtual position of the hand. The object's pose is determined considering the forces acting on the virtual object. Dampers are also allowed in these connections, and both the cases of a serial and a parallel connection have been considered. A virtual grasp matrix is introduced for a better understanding of some proprieties of the grasp, both virtual and real. The dynamics of the system is expressed towards the virtual object and the proposed impedance control law has been applied.

2.3.10 Kinematic control

In order to achieve the desired motion of the manipulated object, the fingers should operate in a coordinated fashion. In the absence of physical interaction between the fingers and the object, simple motion synchronization shall be ensured. On the other hand, the execution of object grasping or manipulation requires controlling also the interaction to ensure the stability of the grasp [72, 82].

From a purely kinematic point of view, an object manipulation task can be assigned in terms of the motion of the fingertips and/or in terms of the desired motion of the manipulated object. The work of the planner is to map the desired task into the corresponding joint trajectories for the fingers and it always requires the solution of an inverse kinematics problem.

Moreover, if the robotic system is equipped with high-gain motion controller at low level, the so called *resolved-velocity* control can be performed, in which the effects of dynamics or disturbances are neglected. In this case, the system is considered as an ideal positioning device and the high controller can act at a velocity level. Since only the kinematics is exploited to derive such control law, often this approach is called *kinematic motion control*, and it is well known in the robotic literature [53, 89], and used also in some manipulation tasks [66].

It is worth noticing that the manipulation system can be redundant also if the single fingers are not: this is due to the presence of the additional degrees of freedom (DOFs) provided by the contact variables. These redundant DOFs can be suitably exploited to satisfy a certain number of secondary tasks, aimed at ensuring grasp stability and manipulation dexterity, besides the main task corresponding to the desired motion of the object. But the kinematic redundancy resolution has not been investigated deeply throughout the literature, since the focus of previous papers seems to be on constrained kinematic control [38, 66], or manipulability analysis [8].

Chapter 3

Visual grasp algorithm

A new method for a fast visual grasp of unknown objects, using a camera mounted on a robot in an eye-in-hand configuration, it is presented in this chapter. This method is composed of an iterative object's surface reconstruction algorithm and of a local grasp planner evolving in a synchronized parallel way.

The reconstruction algorithm makes use of images taken by a camera carried by a robot. First, a rough estimation of the object position and dimensions is performed through the use of a new preshaping technique, then an elastic reconstruction ellipsoidal surface is virtually placed around the object and the fingers of the robotic hand are suitably placed on it. Afterwards, the reconstruction surface iteratively shrinks towards the object's center of mass until some parts of the surface intercept the object's visual hull, dragging out the fingers attached on it. Between two steps of the reconstruction process, the grasp planner moves the fingers according to some grasp quality measures, floating on the current reconstruction surface at an imposed (security) distance.

The fingers keep moving until a local optimum with respect to the chosen grasp quality measure has been reached, and then a new estimation of the surface of the object provided by the reconstruction process is considered. Moreover, a controller should ensure that the motion of the fingers follows the desired trajectories guided by the indices for the grasp. The whole process ends when the object is completely reconstructed and the grasp planner performs the last evaluation of the grasp quality measure, finding the final configuration of the hand and performing the execution of the grasp.

Quality measures considering both hand and grasp proprieties are adopted: the directions of the finger motion leading toward grasp configurations which are not physically reachable, or causing collisions or loss of hand's manipulability, are discarded. A new heuristic grasp quality method has been here proposed, but it is worth noticing that many other quality measures may be chosen in substitution of the proposed one, without effecting the general framework.

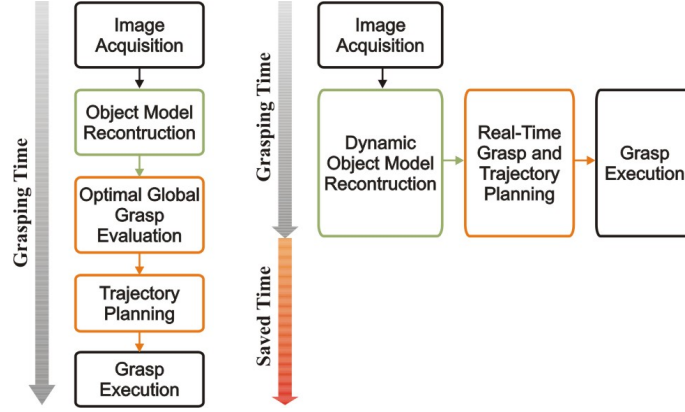


Figure 3.1: Classical serial method vs. proposed parallel approach.

Simulation results and experiments are presented to show the performance of the visual grasp algorithm. The results presented in this chapter can be also found in the following papers [56, 57, 58, 59, 61].

3.1 Description of the visual grasp algorithm

A typical approach for the grasp of unknown objects, here called *serial approach*, mainly consists of two stages: first, the object is completely reconstructed, and then the evaluation of the optimal grasp starts under a selected global criterion, as showed in the Figure 3.1. This approach gives the best results in terms of the quality of the grasp, since the evaluation is made in a global way. However, the total execution time is given by the sum of the time due to the reconstruction of the object and the time due to the synthesis and planning of the grasp.

Although the modern technologies allow a fast reconstruction of the object, the investigation of all the possible combinations of points for the grasp or of the set of surfaces which approximate the object (depending on the reconstruction method adopted) could generally require a considerable amount of time. Obviously, this drawback is irrelevant for off-line applications, but it could be unsuitable for real-time grasping, if powerful hardware is unavailable.

The proposed method, here referred as *parallel approach*, may represent a valid alternative in such cases, where the total computational time is given by the slower between the reconstruction and the planning stage (see Figure 3.1). As a matter of fact, these two parallel processes are independent and they can be allocated under different computational resources. However, the achieved final grasp is optimal only in local sense.

The block diagram in the Figure 3.2 shows the data flow and the main elab-

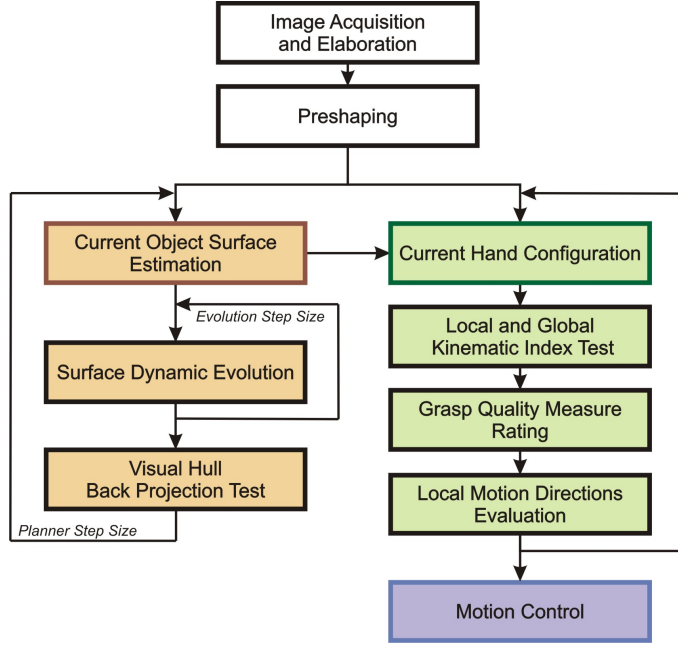


Figure 3.2: The block diagram of the proposed visual grasp algorithm.

oration steps of the proposed visual grasp algorithm. The elaboration processes may be arranged into four main groups, namely:

- *Acquisition of the images and preshaping.*
- *Reconstruction algorithm of the object's surface .*
- *Local planner of the grasp.*
- *Kinematic motion control.*

During the first stage a set of n_{img} images is acquired, and the silhouettes of the object for each image are hence evaluated. Then, the object's center of mass, assuming a homogeneous mass distribution, is estimated using a least-squares triangulation method. More details will be given in the following sections.

The preshaping algorithm gives the dimension of the initial *reconstruction surface*, with elliptical shape, and it is centered at the estimated center of mass of the object. Further, the initial grasp configuration of the hand is evaluated, and it depends on the initial reconstruction surface.

After this preshaping stage, both the object's model reconstruction process and the planner of the grasp start in parallel and cooperate to the final goal. In

particular, as shown in Figure 3.2, the reconstruction algorithm updates in real-time the estimation of the current reconstructed surface of the object, while the motion controller, on the basis of the current estimation, computes the trajectories of the fingertips toward a (local) optimal configuration for the grasp.

At the end, the kinematic motion controller should ensure that the trajectories for the fingertips are correctly followed by the multi-fingered robotic hand: this can be achieved through the use of a robotic system with high gains controllers at the low level, in order to consider the system as an ideal positioning device and in order to perform the so called *resolved-velocity control*.

The assumptions made throughout this work are namely:

- An eye-in-hand configuration with a calibrated camera is available for the reconstruction stage.
- The observed object has to be fixed in the space during image acquisition.
- The observed object has to be distinguishable with respect to the background and other objects.
- From a topological point of view, the observed object must be an orientated surface with genus 0 – in rough words, the object must have no inner hole.
- Any multi-fingered hand suitable for fine manipulation tasks can be considered.

The remaining part of the chapter will follow the outline given by the diagram in Figure 3.2 and already briefly explained before.

3.2 Preparation stages

The preparation stages of the algorithm start with a detection algorithm which is basically based on a classic bob analysis, in order to evaluate the presence of an object in the field of view of the camera. Successively, by holding the optical axis perpendicular to the plane where the object has been detected, the camera is moved until the optical axis intercepts the estimated centroid of the object. At the end of this stage, the camera is exactly over the unknown object and the process of the image acquisitions is ready to start.

3.2.1 Acquisition of the images

The stations for the acquisition of the images are generally chosen as illustrated in Figure 3.3, while the concerning process is carried out as follows:

1. an image is acquired from the top of the object;

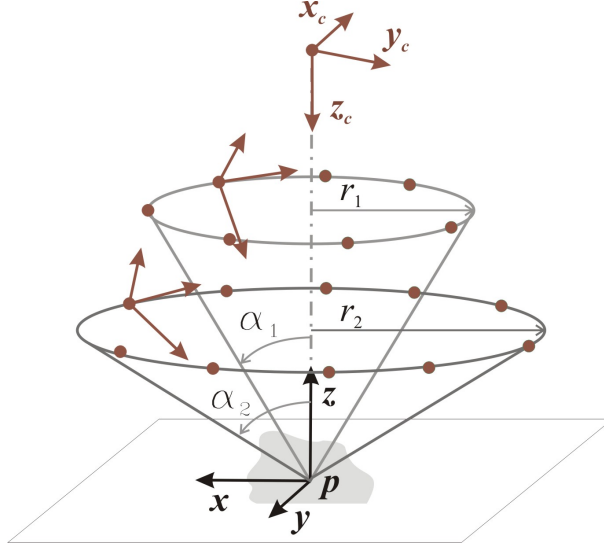


Figure 3.3: Camera stations (bullets) and trajectories of the camera during the acquisition of the images.

2. a subset of n_1 images is taken from camera stations equally distributed over a circular path of radius r_1 , with the optical axis of the camera pointing to the estimated center of the object and forming an angle α_1 with respect to the revolution axis \mathbf{z} ;
3. a subset of n_2 images is acquired as in 2), but using a radius r_2 and an angle α_2 .

In the following, the total number of acquired images will be denoted as $n_{img} = n_1 + n_2 + 1$. It is worth noticing that not all the previous steps should be performed to carry out the process of acquisition of the images, but also a subset of the previous steps can be accomplished.

3.2.2 Elaboration of the images

Once the n_{img} images have been acquired on these circular trajectories, the process of elaboration of these images can start. First, the silhouettes of the images need to be determined: to this aim, a simple binarization process, with a self-tuned threshold, has been employed. Then, a process of a binary dilation and erosion may be necessary to reduce the effects of the image noise and of a not perfect illumination. Moreover, another enhancement of the images can be considered both in spatial and in frequency domains, with the goal to reduce noise and disturbances such as the presence of shadows in the views.

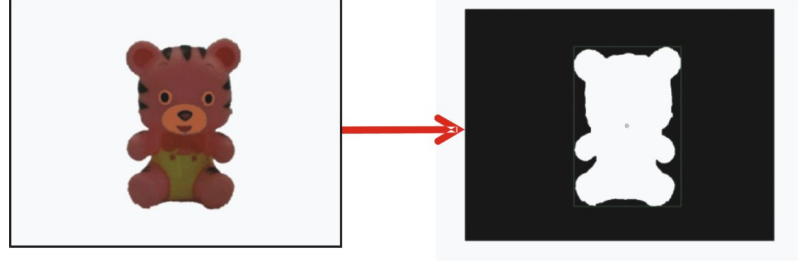


Figure 3.4: Example of the image processing technique employed to extract the silhouette and the bounding box from an acquired image of the observed object.

Once obtained the silhouette for a given image, it is straightforward to determine the corresponding bounding box just considering the smallest rectangle which contains the whole silhouette.

Figure 3.4 shows an acquired image of the unknown object and the resulting silhouette and bounding box after the explained process. Obviously, this process must be performed for all the n_{img} acquired images.

3.2.3 Preshape of the object

The proposed preshaping method starts from a concept presented in [29], where the evaluation of a rough shape estimation is made by a linear programming technique.

Starting from the silhouettes of the acquired n_{img} images and the relative bounding boxes, for each image, the four planes of the Cartesian space containing the origin of the camera frame and two adjacent vertices of the corresponding silhouette bounding box in the image plane are considered, resulting in $4n_{img}$ Cartesian planes. Each plane splits the Cartesian space into two regions, one of which contains the visual hull. The intersections of all these planes create a polyhedron \mathcal{P} which contains the object visual hull, or in other words, is a polyhedral overestimation of this last.

The vertices of this polyhedron can be quickly computed by solving a linear programming problem. Since each side of each bounding box is associated with a plane, if the normal unit vector to the plane is chosen pointing outwards with respect to the interior side of the bounding box, the inner space created by the considered planes for a given image is represented by the following set of inequalities:

$$A_i \mathbf{x} \leq \mathbf{d}_i,$$

where subscript i denotes the i -th image, with $i = 1, \dots, n_{img}$, A_i is a 4×3 matrix whose rows are the transpose of the normal unit vectors, and \mathbf{d}_i is a 4×1 vector whose elements define uniquely the positions of the planes in the space.

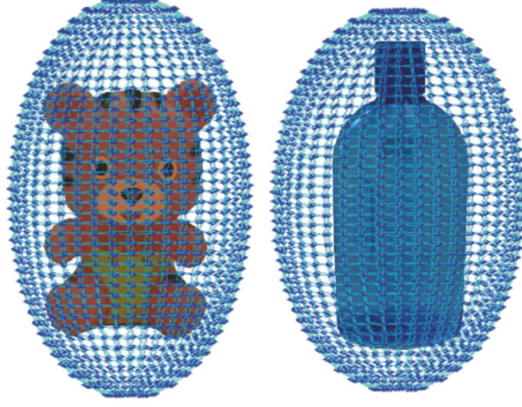


Figure 3.5: Examples of the preshaped elliptical reconstruction surfaces.

Stacking all the A_i and \mathbf{d}_i in the matrices A and \mathbf{d} , the inner space of the polyhedron \mathcal{P} is represented by the following set of inequalities:

$$A\mathbf{x} \leq \mathbf{d}. \quad (3.1)$$

Considering (3.1) as a set of constraints in a minimization problem, the vertices of the corresponding polyhedron are the so called *basic feasible solutions*, whose computation is well known in literature. It is worth noticing that, since the problem has been formulated as a linear programming problem, the computational time is very short and it depends only on the number n_{img} of images.

Once all the n_v vertices $\mathbf{x}_v = [x_{v_x} \ x_{v_y} \ x_{v_z}]^T$ of the polyhedron \mathcal{P} have been computed, the central moments can be evaluated as:

$$\mu_{i,j,k} = \sum_{\mathbf{x}_v \in \mathcal{P}} (x_{v_x} - \bar{x}_{v_x})^i (x_{v_y} - \bar{x}_{v_y})^j (x_{v_z} - \bar{x}_{v_z})^k,$$

where $\bar{\mathbf{x}}_v = [\bar{x}_{v_x} \ \bar{x}_{v_y} \ \bar{x}_{v_z}]^T = \frac{1}{n_v} \sum_{i=1}^{n_v} \mathbf{x}_{v_i}$.

Finally, a pseudo-inertia tensor of the polyhedron can be defined as:

$$I = \begin{bmatrix} \mu_{2,0,0} & \mu_{1,1,0} & \mu_{1,0,1} \\ \mu_{1,1,0} & \mu_{0,2,0} & \mu_{0,1,1} \\ \mu_{1,0,1} & \mu_{0,1,1} & \mu_{0,0,2} \end{bmatrix},$$

where its eigenvalues and eigenvectors define the principal axes of inertia of an ellipsoid, suitably enlarged ensuring object wrapping (see Figure 3.5), which is employed as the initial shape of the reconstruction surface.

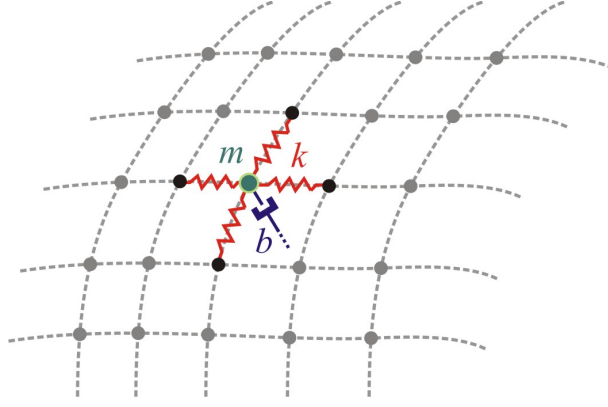


Figure 3.6: Cross network topology of the reconstruction surface with the virtual mass, springs and spatial damper of the i -th sample points.

3.3 Reconstruction algorithm

As described in the previous sections, from the set of n_{img} silhouettes of the object, an elliptical initial reconstruction surface is virtually generated, placed around the object and sampled by n_s reconstruction points.

A virtual mass is associated to each sample point, and four links are imposed with the closest cross points with springs, resulting in a cross reticular topology for the reconstruction surface (see Figure 3.6). The two poles of the ellipsoid are connected with all the masses of the nearest parallel of the resulting ellipsoidal reticulum. A spatial damper has been also considered for each mass.

Each parallel of the ellipsoid should have the same n_m number of points, corresponding to the number of meridians, allowing the construction of a fully linked cross reticulum. In other words, for each mass, the existence of a couple of corresponding points on the closest parallels of the grid is guaranteed. Without loss of generality, the number of parallels n_p is chosen equal to the number of meridians $n_p = n_m = \sqrt{n_s - 2}$. To avoid that the parallels nearest to the poles determine an unnecessary initial thickening of sample points around the poles, a suitable angular distribution of the parallels has been carried out, reducing (augmenting) the density of the parallels near the poles (equator).

The model of the system, defining the motion of each sample point of the reconstruction surface, is defined by the following dynamic equations:

$$m\ddot{\mathbf{x}}_{i,j} + b\dot{\mathbf{x}}_{i,j} + k(4\mathbf{x}_{i,j} - \mathbf{x}_{i-1,j} - \mathbf{x}_{i,j+1} - \mathbf{x}_{i+1,j} - \mathbf{x}_{i,j-1}) = \mathbf{f}_{i,j},$$

for $i = 1, \dots, n_m$ and $j = 1, \dots, n_p$, denoting with $\mathbf{x}_{i,j}$ the position in the workspace of the sampling point at the intersection of the i -th meridian with

the j -th parallel – reticular position (i, j) –, with m , k , and b the mass associated to the point, the constant spring linking the point with its nearest four points of the cross of the reticulum, and the constant spatial damper, respectively. Notice that subscript $i = j = 0$ ($i = n_m + 1$ and $j = n_p + 1$) for the representation of the four nearest points in the reticulum corresponds to $i = n_m$ and $j = n_p$ ($i = j = 1$), respectively.

The two poles have to be treated separately due to their topological peculiarity. For these points, the previous model becomes

$$m\ddot{\mathbf{x}}_{nt} + b\dot{\mathbf{x}}_{nt} + k(n_m\mathbf{x}_{nt} - \sum_{j=1}^{n_m} \mathbf{x}_{1,j}) = \mathbf{f}_{nt}$$

for the north pole, and

$$m\ddot{\mathbf{x}}_{st} + b\dot{\mathbf{x}}_{st} + k(n_m\mathbf{x}_{st} - \sum_{j=1}^{n_m} \mathbf{x}_{n_p,j}) = \mathbf{f}_{st}$$

for the south pole, where the subscripts nt and st indicate quantities refereed to the north and south pole, respectively.

The term $\mathbf{f}_{i,j}$ is the external force acting on the mass associated to the sample point (i, j) , attractive with respect to the border of the visual hull, which is progressively reduced once the corresponding point comes in or out from the visual hull:

$$\mathbf{f}_{i,j} = \alpha_{i,j}(t_{i,j})F_a\mathbf{n}_{i,j},$$

where $\mathbf{n}_{i,j}$ is the unit vector pointing from the current point (i, j) to the estimated centroid of the object, that defines the direction of the force, and $\alpha_{i,j}(t_{i,j})F_a$ is the amplitude of the force. In particular, F_a is the maximum force module and $\alpha_{i,j}(t_{i,j}) \in (-1, 1]$ is a discrete numerical sequence of scale factors which is defined as follows:

$$\alpha_{i,j}(t_{i,j}) = -\epsilon\alpha_{i,j}(t_{i,j} - 1)$$

where $\epsilon \in (0, 1)$, $\alpha_{i,j}(0) = 1$, and $t_{i,j} = 0, \dots, \infty$ is a discrete step time incremented every time the point (i, j) comes in or out of the visual hull.

The stability of the system, for any non-trivial initial condition of the ellipsoid, leads the reconstruction elastic surface to shrink toward its center of mass until the visual hull is intersected. The dynamic evolution of the system reaches the equilibrium when the shape of the surface produces a dynamic equilibrium between the elastic forces generated by the grid and the repulsive forces depending on the contours of the visual hull. The result is that the initial elastic reconstruction surface shapes itself on the unknown object.

From a performance point of view, the accuracy of the reconstruction process depends on:

- the number of views;
- the distribution of the observation stations;
- the density of points of the reconstruction ellipsoid.

On the other hand, the computational time of the algorithm increases if n_{img} and/or n_s are increased. However, considering that the final goal of the process is the grasp of the object and not the model reconstruction, which can be considered as a secondary outcome of the proposed method, the accuracy of the reconstruction process needs only to be adequate for the requirements of the grasp planner algorithm, as already stated in [9].

3.4 Local grasp planner

The current estimation of the object's surface is stored in a proper buffer, which is continuously updated during the dynamic evolution of the elastic surface, and it is employed by the local grasp planner for updating the trajectories of the fingertips. The local planner for the grasp, in accordance with the current reconstructed object's surface, generates the fingertips trajectories on the basis of suitable quality indices of the grasp, but keeping a fixed *floating safety distance* δ_f between the fingertips and the corresponding reconstruction points, along the outgoing normal to the surface. The distance is exploited like a security parameter to avoid undesired collisions between the fingers and the object before the final grasp. See Figure 3.7 for an illustration of the visual grasp concept.

Namely, starting from an initial configuration of the grasp, the planner generates the motion of the fingertips from the current position to a new set of points upon the update surface, according to the selected measure index for the grasp, until no improvements in the quality of the grasp are reached. This new configuration of the contact points will be the new initial configuration of the grasp for the next iteration of the local grasp planner. The process ends when the object reconstruction algorithm reaches an equilibrium and the planner computes the final grasp configuration.

It is worth noticing that any suitable quality index for the grasp can be used in this proposed framework about visual grasp: the index should just ensure that it can be applied to discretized surfaces of the objects. In this thesis, for the sake of clarity, an index that associates a virtual field of forces at each contact point is proposed, while another index is proposed in the Appendix A.

3.4.1 Initial configuration of the grasp

Depending on the object's shape, the initial ellipsoid may have one axis bigger than others, one axis smaller than others, or all axes of a similar dimension. For

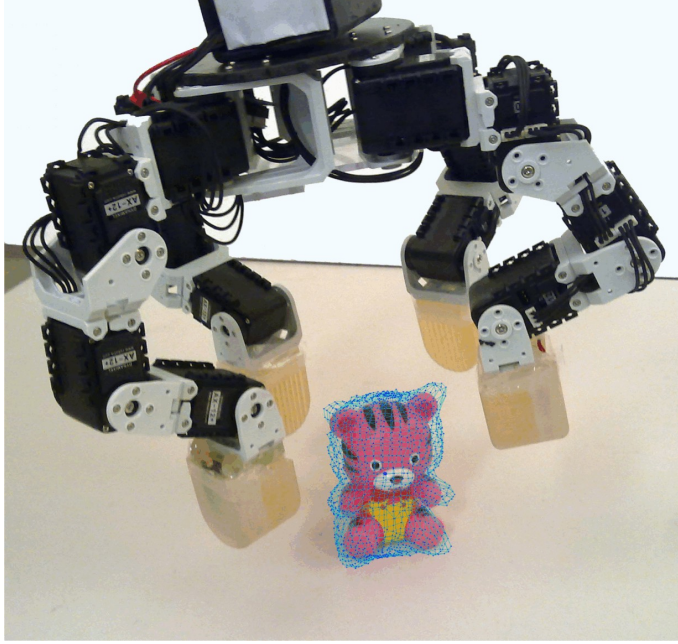
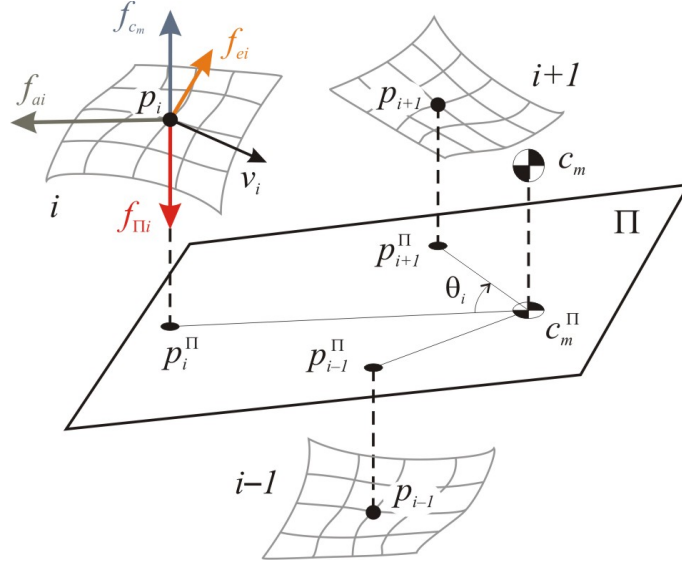


Figure 3.7: Concept of the proposed visual grasp.

all these cases, a good choice for the configuration of the initial grasp depends also on the task to accomplish (e.g. pick-and-place, manipulation, assembling, etc.), on the type of grasp to perform (firm or fine), on the environmental constraints (e.g. the ground plane), and on the hand's kinematics and the number of fingers. In this thesis, for simplicity reasons and considering the previous assumptions of fine manipulation, the initial grasp configuration is chosen as an equilateral grasp in a plane parallel to the two minor axes of the ellipsoid, when it is reachable with respect to the hand and environmental constraints— see [22] for precision grasps in the circular case.

3.4.2 Proposed quality measure for the grasp

In this thesis, planar grasps in the 3D space are considered, assuming that the moments and transversal forces acting on the object can be neglected. In particular, the desired optimal grasp is characterized by having all the contact points lying on the same grasping plane in an equilateral configuration [64, 95]. This choice guarantees force closure for a large number of situations and it simplifies the computation of good grasps, although it may exclude a number of grasp configurations that can be more effective. Moreover, the area of the grasp polygon,


 Figure 3.8: Force field for the i -th contact point.

resulting from the projection of the contact points on the grasp plane, has to be maximized in order to improve the quality of the grasp with respect to possible external moments normal to the grip plane [64]. Finally, if it is required by the desired application, it can be also imposed that the center of mass of the current reconstruction surface (that is equivalent to the reconstructed object's center of mass at the end of the process) has to be contained in the current grasping plane, enhancing the minimization of gravity and inertial effects during manipulation tasks [26, 79].

Motion field of forces

In order to accomplish what said before, a field of forces defined as the sum of suitable force contributions is associated at each contact point. First, the interpolating plane Π of the current contact points — i.e., the plane which minimizes the distance from all the contact points —, and the projections \mathbf{p}_i^Π of the contact point \mathbf{p}_i on Π , with $i = 1, \dots, n_f$, are computed (see Figure 3.8), where n_f is the number of the fingers of the robotic hand. Then, the force associated to the i -th contact point is:

$$\mathbf{f}_i = \mathbf{f}_{\Pi i} + \mathbf{f}_{c_m} + \mathbf{f}_{ei} + \mathbf{f}_{ai} + \mathbf{f}_{bi}, \quad (3.2)$$

where each contribution of force, with reference to Fig. 3.8, is defined as follows:

- $\mathbf{f}_{\Pi i} = k_\Pi (\mathbf{p}_i^\Pi - \mathbf{p}_i)$ is the force which moves \mathbf{p}_i to \mathbf{p}_i^Π , so that all the contact points belong to the same grasp plane;

- $\mathbf{f}_{c_m} = k_{c_m} (\mathbf{c}_m - \mathbf{c}_m^\Pi)$ is the force, equal for all the contact points, which attracts the grasp plane Π to \mathbf{c}_m , where \mathbf{c}_m is the center of mass of the current reconstruction surface and \mathbf{c}_m^Π is the projection of the center of mass on the interpolating plane;
- $\mathbf{f}_{e_i} = k_e (\theta_i - \frac{2\pi}{n_f}) \mathbf{t}_i$ is the tangential force which is in charge of producing an equilateral grasp configuration, where θ_i is the angle between the vectors $\mathbf{p}_i^\Pi - \mathbf{c}_m^\Pi$ and $\mathbf{p}_j^\Pi - \mathbf{c}_m^\Pi$, with $j = i + 1$ for $i = 1, \dots, n_f - 1$, and $j = 1$ for $i = n_f$, and \mathbf{t}_i is the tangential unit vector normal to $\mathbf{c}_m^\Pi - \mathbf{p}_i^\Pi$, lying on Π and pointing toward \mathbf{p}_j^Π ;
- $\mathbf{f}_{a_i} = k_a (\mathbf{p}_i^\Pi - \mathbf{c}_m^\Pi) / \|\mathbf{p}_i^\Pi - \mathbf{c}_m^\Pi\|$ is a force component which tends to enlarge the area of the grasp polygon;
- \mathbf{f}_{b_i} represents the kinematic barrier force, depending on the local and global kinematic indices, described in the next subsection.

The parameters k_Π, k_{c_m}, k_e, k_a are all positive constant coefficients, suitably designed to weigh the single force contributions with respect to the requirements of the single situations and/or tasks to accomplish.

Kinematic barrier of forces

The kinematic barrier of forces \mathbf{f}_{b_i} for the i -th contact point is aimed at avoiding the motion of the finger along directions that cause the reaching of joint limits, joint or hand singularities, and collisions between fingers or with the palm. In detail, the barrier of forces is equal to

$$\mathbf{f}_{b_i} = \mathbf{f}_{j_i} + \mathbf{f}_{s_i} + \mathbf{f}_{c_i},$$

where each term is related to one of the neighbor points of the contour, directed from the corresponding contour point towards the actual contact point:

- \mathbf{f}_{j_i} is zero when the finger joint positions are far from their limits, while it quickly increases its magnitude, with a hyperbolic law, when one or more joint limits are approached at least for one of the contour points. Therefore, the force \mathbf{f}_{j_i} is in charge to move the contact point far from unreachable positions.
- \mathbf{f}_{s_i} is zero when the configuration of the finger is far from kinematic singularities, while it quickly increases, with a hyperbolic law, when a kinematic singularity is approached. Therefore, \mathbf{f}_{s_i} represents a force that is repulsive with respect to the directions leading to finger singularities.
- \mathbf{f}_{c_i} is zero when the fingers are far from each others and from the palm, while its magnitude is increased when a safety distance is violated.

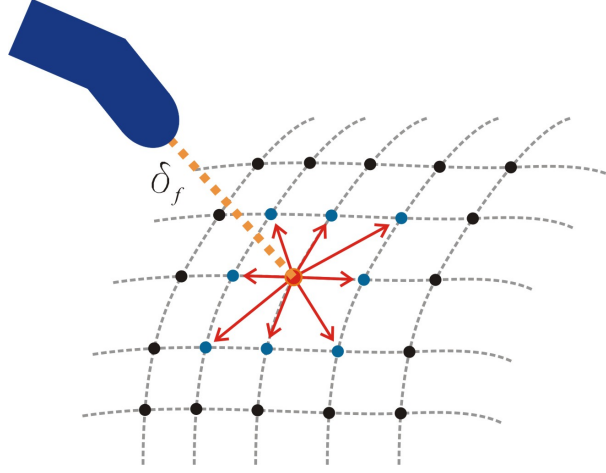


Figure 3.9: Contour of neighbor points of the current target grasp point and possible direction of movements of the associated finger.

Moreover, the barrier forces can be also employed to cope with environmental constraints, e.g. object ground plane or other surrounding objects.

Evaluation of the quality of the grasp

The force \mathbf{f}_i is projected onto the tangential plane to the current reconstruction surface at the contact point i , determining the direction of motion for the i -th contact point:

$$\mathbf{f}'_i = \mathbf{f}_i - (\mathbf{f}_i^T \mathbf{v}_i) \mathbf{v}_i,$$

where \mathbf{v}_i is the unit normal vector to the reconstruction surface at the point \mathbf{p}_i .

The direction of \mathbf{f}'_i individuates one of the points of the surface close to the current one, as shown in Figure 3.9, and it is employed by the planner to produce the floating motion of the finger. When $\|\mathbf{f}'_i\|$ is higher than a given threshold σ_f , the current grasp configuration changes according to the directions of \mathbf{f}'_i . The choice of σ_f means that forces \mathbf{f}'_i whose norm is under this threshold can be neglected, and when this happens for all the contact points, then the reached configuration is the (local) optimal grasp for the current iteration. Obviously σ_f affects both the accuracy of the grasp solution and the computational time, determining the number of iterations required to converge to the local optimum, and thus it must be suitably tuned considering this trade-off.

It is worth noticing that when the sum of each contribution in (3.2) for a finger results in a zero force field, the corresponding contact point does not change its position in the actual step of the current iteration of the planner stage.

3.5 Generation of the fingers trajectories and motion controller

The local grasp planner produces a sequence of intermediate target grasp configurations at each iteration of the object's reconstruction algorithm, which ends with the optimal grasp configuration (in a local sense). The intermediate configurations are used to generate the paths for the fingertips.

Namely, the sequence of intermediate configurations is suitably filtered by a spatial low-pass filter in order to achieve a smooth path for the fingers on the object surface. Notice that only the final configuration needs to be reached exactly, while the intermediate configurations can be considered as via points for the generation of the trajectories of the fingers, and that can be computed in real-time with a one step delay.

With respect to the smooth paths through the points of the filtered configurations, the actual paths of the fingers generated by the trajectory planner keep a distance δ_f along the normal to the surface. When the final configuration is reached, the safety distance is progressively reduced to zero, producing the desired grasp action, with directions of grasp perpendicular to the object reconstructed surface.

A kinematic control is used to allow a correct tracking of the trajectories given by the planner. In particular, a closed-loop inverse-kinematic algorithm [89] has been exploited to reach such a goal. This method requires as input the desired position for each finger, and as output it gives the joints velocity for each finger. High gain low-level controllers are necessary to physically accomplish the task. Moreover, an interaction control (e.g. an impedance control) can be used to touch the object with a desired dynamic, and a force optimization algorithm [12] could be used for a proper distribution of the grasp forces.

3.6 Simulations and experiments

The proposed method about visual grasp has been experimentally tested on different real objects considering a different number of fingers of the available robotic hand of Figure 3.7. Obviously, since it is not an anthropomorphic hand, a human-like grasp here means that it must be stable and the hand should be in a feasible and dexterous configuration. In the following, the results for the objects shown in Figure 3.5, a teddy-bear and a little bottle, are presented.

Images in a number equal to $n_{img} = 13$ have been taken for both the objects by a common webcam mounted inside the palm of the hand in Figure 3.7. The reconstruction surface is sampled with $n_s = 1500$ points, while the reconstruction parameters have been chosen as: $m = 10^{-3}$ kg, $k = 0.3 \cdot 10^{-3}$ N/m, $b = 0.09 \cdot 10^{-3}$ Ns/m, and $F_a = 5$ N. For the grasp planner, $k_{\Pi}, k_{c_m}, k_e, k_a$ have been chosen

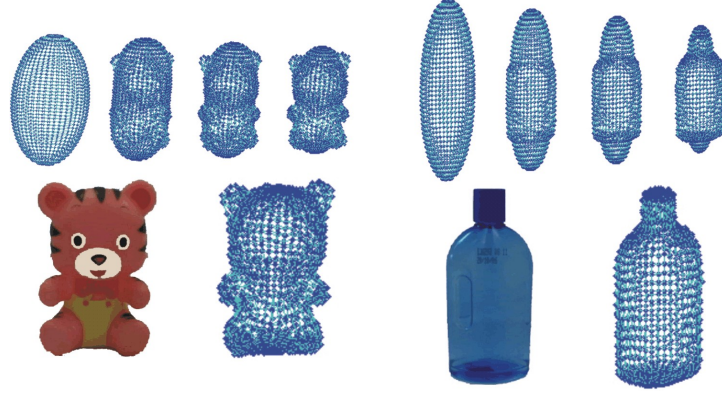


Figure 3.10: Steps of the process about reconstruction of the object's surface: on the left a teddy bear, on the right a bottle.

all equal to 1, in order to have an equivalent weight for all the contributions, while the threshold σ_f has been tuned to a value of 0.002 N. The floating security distance δ_f has been set to 2 cm, which is deliberately a huge value for a better visualization of the trajectories. The computational time for the whole process is about 1.5 s on a Pentium 1.7 GHz: in particular, the planner stage is the slower one, since the stage about the reconstruction of the object employs only 80 ms, for $n_s = 1500$ points, to accomplish its job.

In Figure 3.10 some intermediate steps of the reconstruction algorithm are shown, while the finger trajectories and the final grasp configurations, respectively for the teddy-bear and for the little bottle, are shown in Fig. 3.11. Both cases of $k_{c_m} = 1$ (left) and $k_{c_m} = 0$ (right) are considered (the bold point represents the position of the object's center of mass of the reconstructed object). In particular, for the case $k_{c_m} = 1$ it is evident that both the grasp planes of the final grasps contain the center of mass of the objects, while for the case $k_{c_m} = 0$ the plane of the final grasp is far from the center of mass to achieve a more extended areas of the grasp polygon.

More in detail, Figure 3.11 shows how the teddy-bear is grasped with three fingers achieving a desirable stable planar equilateral grasp (120° apart) for both cases of $k_{c_m} = 1$ and $k_{c_m} = 0$. The yellow lines represent the sequence of reconstruction points selected by the planner during the evolution of the reconstruction surface. The green lines represent the trajectories that the planner generates for the fingertips after spatial filtering and considering the safety distance. Finally, the red lines show the last part of the grasp trajectory, when the safety distance is progressively reduced achieving a perpendicular approach to the object's surface. It is worth noticing that in the case $k_{c_m} = 0$ the fingers aim to reach a larger

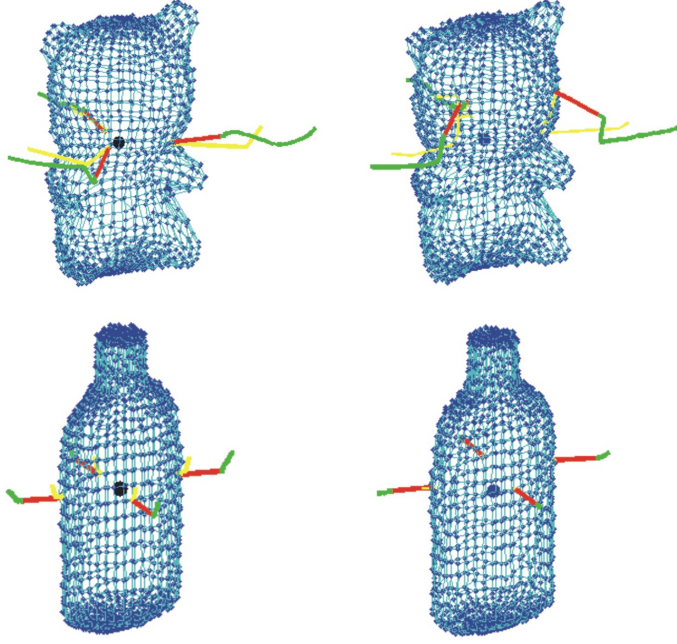


Figure 3.11: Trajectories of the fingers generated by the local grasp planner (green: approach, red: grasp) and the corresponding sequence of floating grasp points achieved during the reconstruction process (yellow) for the two objects, both evaluated with $k_{cm} = 1$ (left) and $k_{cm} = 0$ (right).

polygon area to grasp since they are not anymore constrained to have the grasp plane containing the center of mass.

For the case of the little bottle, four fingers of the hand have been considered. The final grasp configuration corresponds to the equilateral best grasp (90° apart) for the object. Moreover, the achieved trajectories are very regular due to the good choice of the initial grasp configuration evaluated by the preshaping module. This result is common when the object is symmetric with respect to one or more axes, and so it is well represented by an ellipsoidal surface. Of course, for the particular shape of the bottle, the results do not change significantly when k_{cm} is set to 0.

Furthermore, to validate the proposed method, a comparison between the results obtained with this approach and those obtained with traditional grasp quality indices has been performed. To make this comparison, the classical serial approach has been implemented. Hence, after the whole reconstruction of the unknown object, a global search of the optimal grasp has been performed according to three well-known quality indices: namely, Q_1 the max-min singular value of the grasp

Table 3.1: Comparison between the numerical evaluations – through the use of some traditional grasp quality measures – of the grasp’s final configurations obtained with the serial approach and the parallel one. For the latter, the final configuration of the grasping points has been found using the proposed quality index.

$k_{c_m} = 1$	Q_1		Q_2		Q_3	
	<i>Local</i>	<i>Global</i>	<i>Local</i>	<i>Global</i>	<i>Local</i>	<i>Global</i>
Bottle	0.016	0.016	0.091	0.101	0.438	0.438
Teddy-Bear	0.292	0.320	0.536	0.690	0.426	0.482
$k_{c_m} = 0$	<i>Local</i>	<i>Global</i>	<i>Local</i>	<i>Global</i>	<i>Local</i>	<i>Global</i>
Bottle	0.019	0.020	0.099	0.112	0.574	0.590
Teddy-Bear	0.267	0.297	0.369	0.413	0.707	0.780

matrix [55], Q_2 the maximum volume of the ellipsoid in the wrench space [55] and Q_3 the largest perturbation wrench that the grasp can resist [32]. Where requested, a frictionless contact assumption has been done. Two further constraints have been included in the global search: grasp configurations which violated hand physical constraints and grasp configurations whose center of grip is far from the object’s center of mass have been both neglected. The latter has been obviously considered only in the case where $k_{c_m} = 1$. Then, once selected for example the Q_1 index, once the global optimal configuration for the grasp has been found with the serial approach and hence evaluated, the final grasp’s configuration obtained with the parallel algorithm and the proposed quality index has been evaluated as well with Q_1 . Of course, this operation has been repeated with Q_2 and Q_3 too.

The results of this comparison are shown in Table 3.1, where it is evident that the performances are very close in the case of the little bottle, while there is a small difference in the case of the teddy-bear. This result could be explained with the fact that the more regular is the object’s surface, the better are the results of the proposed local approach with respect to the global ones. Moreover, the evaluated final contact points in the global and in the local case are very close to each other, especially in the case of symmetric objects. These statements are confirmed by experiments performed with other objects, such as a cylinder, an Easter egg, a glass, which are here omitted for brevity.

Chapter 4

CLIK algorithm for dexterous manipulation

A kinematic model for object manipulation using a multi-fingered robotic hand has been derived in this chapter. The model allows the object's pose to be computed from the joint variables of each finger, called *active joints*, as well as from a set of contact variables modeled as *passive joints* [66]. Suitable conditions have been derived, ensuring that a given motion can be imposed to the object using only the active joints. A closed-loop inverse kinematics algorithm (CLIK [89]) has been re-arranged in this context, in order to compute finger contact variables, once given the desired trajectory of the object. Both the schemes based on the transpose and on the inverse of the Jacobian matrix can be adopted in the proposed framework.

Further, as already sentenced in Chapter 2, the manipulation system can be redundant also if the single fingers are not: this is due to the presence of the additional degrees of freedom (DOFs) provided by the contact variables. These redundant DOFs can be suitably exploited to satisfy a certain number of secondary tasks, aimed at ensuring grasp stability and manipulation dexterity, besides the main task corresponding to the desired motion of the object.

Simulations are presented to show the performance of the proposed re-arranged CLIK and the results presented in this chapter can be also found in the following papers [60, 62].

4.1 Kinematics of objet and fingers

Consider a robotic hand composed by n_f rigid fingers, numbered from 1 to n_f , holding a rigid object, and let \mathbf{q}_i denote the joint vector of finger i , with n_i components. To derive the kinematic mapping between the joint variables of the fingers and the pose (position and orientation) of the object, it is useful introducing an *object frame* Σ_o attached to the object, usually chosen with the origin in the

object's center of mass. The pose of Σ_o with respect to a *base frame* Σ_b fixed to the hand (also known as *hand frame*) can be represented by the (4×4) homogeneous transformation matrix

$$T_o = \begin{bmatrix} \mathbf{R}_o & \mathbf{o}_o \\ \mathbf{0}^T & 1 \end{bmatrix},$$

where \mathbf{R}_o is the (3×3) rotation matrix, \mathbf{o}_o is the (3×1) position vector of the origin of Σ_o with respect to the base frame, while $\mathbf{0}$ denotes the (3×1) null vector. The velocity of Σ_o with respect to the base frame can be represented by the (6×1) twist vector $\mathbf{v}_o^T = [\dot{\mathbf{o}}_o^T \ \boldsymbol{\omega}_o^T]^T$, where $\boldsymbol{\omega}_o$ is the angular velocity such that $\dot{\mathbf{R}}_o = \mathbf{S}(\boldsymbol{\omega}_o)\mathbf{R}_o$, with $\mathbf{S}(\cdot)$ the skew-symmetric operator representing the vector product.

Assuming that each finger has only one contact point with the object, it is useful introducing a *finger frame* Σ_{f_i} attached to the i -th fingertip ($i = 1, \dots, n_f$) and with the origin \mathbf{o}_{f_i} at the point of contact. The pose of Σ_{f_i} with respect to the base frame can be computed on the basis of the kinematics of the finger as $T_{f_i} = T_{f_i}(\mathbf{q}_i)$, while the velocity can be expressed as

$$\mathbf{v}_{f_i} = \begin{bmatrix} \dot{\mathbf{o}}_{f_i} \\ \boldsymbol{\omega}_{f_i} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{P_i} \\ \mathbf{J}_{O_i} \end{bmatrix} \dot{\mathbf{q}}_i = \mathbf{J}_i(\mathbf{q}_i) \dot{\mathbf{q}}_i, \quad (4.1)$$

where \mathbf{J}_i is the $(6 \times n_i)$ Jacobian of the finger, while \mathbf{J}_{P_i} and \mathbf{J}_{O_i} are $(3 \times n_i)$ matrices known as position Jacobian and orientation Jacobian respectively.

4.2 Contact kinematics

Grasping situations may involve moving rather than fixed contacts: often, both the object and the robotic fingers are smooth surfaces, and manipulation involves rolling and/or sliding of the fingertips on the object's surface, depending on the contact type. If the fingers and object's shapes are completely known, the contact kinematics can be described introducing contact coordinates defined on the basis of a suitable parametrization of the surfaces [65, 67].

To gain insight into the kinematics of contact, it is assumed that

- the fingertips are sharp so that the contact point \mathbf{o}_{f_i} of each finger is fixed and coincides with (or can be approximated by) the position of the fingertip;
- the object is a smooth surface (see, e.g., [20]).

Let Σ_{c_i} be the *contact frame* attached to the object and with the origin at the contact point \mathbf{o}_{c_i} . Notice that, instantaneously, the object contact point \mathbf{o}_{c_i} and the finger contact point \mathbf{o}_{f_i} are coincident. One of the axes of Σ_{c_i} , e.g., the Z axis, is assumed to be normal to the tangent plane to the object's surface at the contact point, and pointing outward the surface of the object.

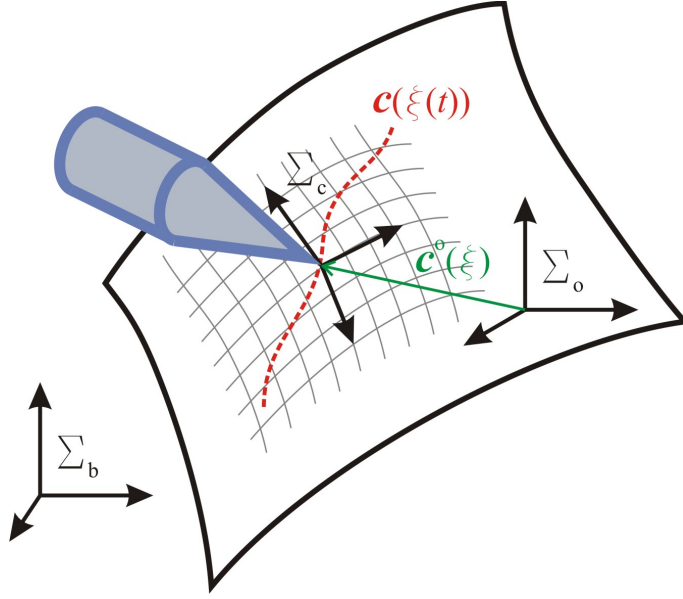


Figure 4.1: Local parametrization of the object's surface with respect to the object frame.

Moreover, assume that, at least locally, the position of the contact point with respect to the object frame $\mathbf{o}_{o,c_i}^o = \mathbf{o}_{c_i}^o - \mathbf{o}_o^o$ can be parameterized in terms of a coordinate chart $\mathbf{c}_i^o : U_i \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$ which maps a point $\boldsymbol{\xi}_i = [u_i \ v_i]^T \in U_i$ into the point $\mathbf{o}_{o,c_i}^o(\boldsymbol{\xi}_i)$ of the surface of the object.

To simplify the notation, for the reminder of this section, index i will be dropped.

In the hypothesis that \mathbf{c}^o is a diffeomorphism and that the coordinate chart is orthogonal and right-handed, contact frame Σ_c can be chosen as a Gauss frame [65], with the rotation matrix \mathbf{R}_c^o computed as

$$\mathbf{R}_c^o(\boldsymbol{\xi}) = \begin{bmatrix} \frac{\mathbf{c}_u^o}{\|\mathbf{c}_u^o\|} & \frac{\mathbf{c}_v^o}{\|\mathbf{c}_v^o\|} & \frac{\mathbf{c}_u^o \times \mathbf{c}_v^o}{\|\mathbf{c}_u^o \times \mathbf{c}_v^o\|} \end{bmatrix},$$

where tangent vectors $\mathbf{c}_u^o = \partial \mathbf{c}^o / \partial u$ and $\mathbf{c}_v^o = \partial \mathbf{c}^o / \partial v$ are orthogonal.

Consider first the contact kinematics from the object point of view. Let $\mathbf{c}^o(\boldsymbol{\xi}(t))$ denote a curve on the surface of the object, with $\boldsymbol{\xi}(t) \in U$ (see Figure 4.1). The corresponding motion of the contact frame with respect to the base frame can be determined as a function of:

- object motion;
- geometric parameters of the object;

- “velocity” of the curve.

Namely, computing the time derivative of the equation $\mathbf{o}_c = \mathbf{o}_o + \mathbf{R}_o \mathbf{c}^o(\boldsymbol{\xi})$, which denotes the position of the object’s contact point in the base frame, yields

$$\dot{\mathbf{o}}_c = \dot{\mathbf{o}}_o + \mathbf{S}(\mathbf{c}^o(\boldsymbol{\xi}))\boldsymbol{\omega}_o + \mathbf{R}_o \frac{\partial \mathbf{c}^o}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}}, \quad (4.2)$$

where the first two terms on the right-hand side specify the velocity contribution due to the motion of the object, while the last term represents the velocity of the motion on the object’s surface relative to the object frame.

On the other hand, for the angular velocity, the following equality holds

$$\boldsymbol{\omega}_c = \boldsymbol{\omega}_o + \mathbf{R}_o \boldsymbol{\omega}_{o,c}^o, \quad (4.3)$$

being $\boldsymbol{\omega}_{o,c}^o$ the angular velocity of the motion of the contact frame relative to the object frame, which can be expressed in the form

$$\boldsymbol{\omega}_{o,c}^o = \mathbf{C}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}, \quad (4.4)$$

with $\mathbf{C}(\boldsymbol{\xi})$ a (3×2) matrix depending on geometric parameters of the surface [67]. Matrix \mathbf{C} is not necessarily full rank; for example, in the case of planar surfaces, this matrix is null.

In view of (4.2), (4.3), (4.4), the velocity of the contact frame can be expressed as

$$\mathbf{v}_c = \begin{bmatrix} \dot{\mathbf{o}}_c \\ \boldsymbol{\omega}_c \end{bmatrix} = \mathbf{G}^T(\boldsymbol{\xi})\mathbf{v}_o + \mathbf{J}_\xi(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}, \quad (4.5)$$

where

$$\mathbf{G}(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{S}(\mathbf{c}(\boldsymbol{\xi})) & \mathbf{I} \end{bmatrix}, \quad \mathbf{J}_\xi(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{R}_o \frac{\partial \mathbf{c}^o}{\partial \boldsymbol{\xi}} \\ \mathbf{R}_o \mathbf{C}(\boldsymbol{\xi}) \end{bmatrix}$$

are respectively (6×6) and (6×2) full rank matrices.

Consider now the contact kinematics from the finger point of view. The contact can be modeled as an unactuated 3-DOF ball and socket kinematic pair centered at the contact point, possibly moving on the surface if sliding is allowed. Hence, the orientation of contact frame Σ_c with respect to finger frame Σ_f can be computed in terms of a suitable parametrization of the ball and socket joint, e.g., Euler angles, axis-angle or unit quaternion. For the purpose of this work, a vector $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3]^T$ of XYZ Euler angles is considered, thus $\mathbf{R}_c^f = \mathbf{R}_c^f(\boldsymbol{\theta})$. In detail, θ_1 and θ_2 parameterize the so-called “swing” motion aligning axis Z of a moving frame to axis Z of the contact frame, while θ_3 corresponds to the “twist” motion about axis Z of the contact frame. Singularities occurs for $\theta_2 = \pm\pi/2$, but they do not correspond to physical singularities of the kinematics pair. Therefore, the angular velocity of Σ_c relative to Σ_f can be expressed as

$$\boldsymbol{\omega}_{f,c}^f = \mathbf{T}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}},$$

where \mathbf{T} is a suitable transformation matrix. In view of the decomposition $\boldsymbol{\omega}_c = \boldsymbol{\omega}_f + \mathbf{R}_f(\mathbf{q})\boldsymbol{\omega}_{f,c}^f$, and from (4.1), the angular velocity of Σ_c can be computed also as a function of finger and contact variables in the form

$$\boldsymbol{\omega}_c = \mathbf{J}_O(\mathbf{q})\dot{\mathbf{q}} + \mathbf{R}_f(\mathbf{q})\mathbf{T}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \quad (4.6)$$

Moreover, since the origins of Σ_c and Σ_f coincide, the following equality holds

$$\dot{\boldsymbol{\omega}}_c = \dot{\boldsymbol{\omega}}_f = \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}}. \quad (4.7)$$

Using (4.6) and (4.7), the velocity of the contact frame can be expressed as

$$\mathbf{v}_c = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}_\theta(\boldsymbol{\theta}, \mathbf{q})\dot{\boldsymbol{\theta}}, \quad (4.8)$$

where \mathbf{J} is the finger Jacobian and

$$\mathbf{J}_\theta = \begin{bmatrix} \mathbf{0} \\ \mathbf{R}_f(\mathbf{q})\mathbf{T}(\boldsymbol{\theta}) \end{bmatrix}$$

is a (6×3) full rank matrix (far from representation singularities).

Hence, from (4.5) and (4.8), the contact kinematics of finger i has the form

$$\mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i + \mathbf{J}_{\eta_i}(\boldsymbol{\eta}_i, \mathbf{q}_i)\dot{\boldsymbol{\eta}}_i = \mathbf{G}_i^T(\boldsymbol{\eta}_i)\mathbf{v}_o, \quad (4.9)$$

where $\boldsymbol{\eta}_i = [\boldsymbol{\xi}_i^T \quad \boldsymbol{\theta}_i^T]^T$ is the vector of contact variables and $\mathbf{J}_{\eta_i} = [-\mathbf{J}_{\xi_i} \quad \mathbf{J}_{\theta_i}]$ is a (6×5) full rank matrix. This equation can be interpreted as the differential kinematics equation of an “extended” finger corresponding to the kinematic chain including the finger joint variables (active joints) and the contact variables (passive joints), from the base frame to the contact frame [66].

It is worth pointing out that equation (4.9) involves all the 6 components of the velocity, differently from the grasping constraint equation usually considered (see, e.g., [67]), which contains only the components of the velocities that are transmitted by the contact. The reason is that the above formulation takes into account also the velocity components not transmitted by the contact i , parameterized by the contact variables and lying in the range space of \mathbf{J}_{η_i} . As a consequence, \mathbf{G}_i is always a full rank matrix.

Planar motions can be analyzed as a particular case of the general 6-DOF motion by rewriting equation (4.9) in terms of 3 components.

Depending on the type of the considered contact, some of the parameters of $\boldsymbol{\xi}_i$ and $\boldsymbol{\theta}_i$ are constant. For the three models usually considered for grasp analysis [82], it is:

- *point contact without friction*: all the parameters may vary (i.e., the finger may slide on the object surface and the orientation of Σ_{f_i} with respect to Σ_{c_i} may vary);

- *hard finger with friction*: vector ξ_i is constant, while vector θ_i may vary (i.e., the finger is not allowed to slide on the object surface, but the orientation of Σ_{f_i} with respect to Σ_{c_i} may vary);
- *soft finger with friction*: vector ξ_i is constant, as well as the last parameter of vector θ_i , corresponding to the rotation about the Z axis of Σ_{c_i} (i.e., the finger is not allowed to slide and to twist about the normal to the surface of the object).

Hence, assuming that the type of contact remains unchanged during task execution, the variable parameters at each contact point are grouped in a $(n_{c_i} \times 1)$ vector η_i of contact variables, with $n_{c_i} \leq 5$.

4.3 Classification of grasps

On the basis of (4.9), it is possible to make a kinematic classification of the grasp [82].

A grasp is said to be *redundant* if the null space of the matrix

$$\tilde{J}_i = \begin{bmatrix} J_i & J_{\eta_i} \end{bmatrix}$$

is non-null, for at least one finger i . In this case, the mapping between the joint variables of the “extended” finger i and the object’s velocity is many to one: motions of active and passive joints of the extended finger are possible when the object is locked. Notice that a single finger could be redundant if the null space of J_i is non-null, i.e., in the case of a kinematically redundant finger; in this case, motions of the active joints are possible when both the passive joints and the object are locked. On the other hand, for the type of contacts considered here (point contact), the null space of J_{η_i} is always null: this implies that motions of the passive joints are not possible when the active joints and the object are locked. In typical situations, the fingers of the robotic hand are not redundant, but the extended fingers may be redundant thanks to the presence of the additional DOFs provided by the passive joints.

A grasp is *indeterminate* if the intersection of the null spaces of $[-J_{\eta_i} \quad G_i^T]$, for all $i = 1, \dots, n_f$, is non-null. In this case, motions of the object and of the passive joints are possible when the active joints of all the fingers are locked. The kinematic indetermination derives from the fact that the motion of the object cannot be completely controlled by finger motions, but depends on the dynamics of the system [67]. An example of indeterminate grasp may be that of a box grasped by two hard-finger opposite contacts: in this case, the box may rotate about the axis connecting the two contact points while the fingers are locked.

It is worth pointing out that, also in the case of redundant and indeterminate grasps, for a given object’s pose and a configuration of the fingers, the value of the contact variables is uniquely determined.

4.4 Kinematic motion control with redundancy resolution

In the case of kinematically determinate and, possibly, redundant grasp, a suitable inverse kinematics algorithm can be adopted to compute the fingers and contact variables corresponding to a desired motion of the object. Hence, this framework can act as a kinematic motion controller to perform the desired object manipulation task.

In detail, in view of (4.9), the CLIK algorithm with redundancy resolution, based on the pseudo-inverse of the Jacobian matrix, is given by equation:

$$\begin{bmatrix} \dot{\mathbf{q}}_i \\ \dot{\boldsymbol{\eta}}_i \end{bmatrix} = \tilde{\mathbf{J}}_i^\dagger(\mathbf{q}_i, \boldsymbol{\eta}_i) \mathbf{G}_i^T(\boldsymbol{\eta}_i) (\mathbf{v}_d + \mathbf{K}_i \mathbf{e}_{o_i}) + \mathbf{N}_{o_i}(\mathbf{q}_i, \boldsymbol{\eta}_i) \boldsymbol{\sigma}_i, \quad (4.10)$$

where $\tilde{\mathbf{J}}_i^\dagger$ is a right (weighted) pseudo-inverse of $\tilde{\mathbf{J}}_i$, \mathbf{e}_{o_i} is a pose error between the desired and the current pose of the object, \mathbf{K}_i is a (6×6) symmetric and positive definite matrix, $\boldsymbol{\sigma}_i$ is a suitable velocity vector corresponding to a secondary task, and

$$\mathbf{N}_{o_i} = \mathbf{I} - \tilde{\mathbf{J}}_i^\dagger \tilde{\mathbf{J}}_i \quad (4.11)$$

is a projector in the null space of the Jacobian matrix. The asymptotic stability of the equilibrium $\mathbf{e}_{o_i} = \mathbf{0}$ for system (4.10) can be easily proven (see [89] for the case of a standard CLIK algorithm).

The computation of the Jacobian pseudo-inverse can be avoided by adopting an alternative CLIK algorithm based on the transpose of the Jacobian matrix, given by equation:

$$\begin{bmatrix} \dot{\mathbf{q}}_i \\ \dot{\boldsymbol{\eta}}_i \end{bmatrix} = \tilde{\mathbf{J}}_i^T(\mathbf{q}_i, \boldsymbol{\eta}_i) \mathbf{G}_i^T(\boldsymbol{\eta}_i) \mathbf{K}_i \mathbf{e}_{o_i} + \mathbf{N}_{o_i}(\mathbf{q}_i, \boldsymbol{\eta}_i) \boldsymbol{\sigma}_i. \quad (4.12)$$

The asymptotic stability of the equilibrium $\mathbf{e}_{o_i} = \mathbf{0}$ for system (4.12) can be easily proven in the case $\mathbf{v}_d = \mathbf{0}$ (see [89] for the case of a standard CLIK algorithm). Notice that, if \mathbf{N}_{o_i} is computed according to (4.11), the computation of the Jacobian pseudo-inverse is not avoided with algorithm (4.12).

In principle, n_f independent CLIK algorithms can be used, one for each finger, all with the same input, namely, the desired object's pose \mathbf{T}_d and velocity \mathbf{v}_d . However, some secondary tasks may involve all the variables of the system at the same time, e.g., those related to quality of the grasp.

Hence, the complete CLIK scheme with redundancy resolution includes n_f decentralized feedback loops, one for each finger, and a centralized feedforward action depending on the whole configuration of the system. This is shown in the block diagram of Figure 4.2 for the scheme based on the pseudo-inverse of the Jacobian matrix, in which $\tilde{\mathbf{q}}_i = [\mathbf{q}_i^T \quad \boldsymbol{\eta}_i^T]^T$.

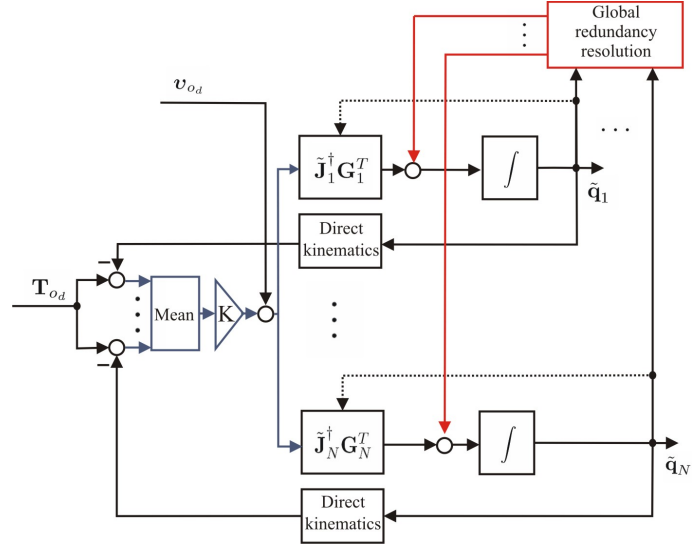


Figure 4.2: Block scheme of the CLIK algorithm with redundancy resolution based on equation (4.10). Notice that $N = n_f$.

Since the system may be highly redundant, multiple tasks could be fulfilled, provided that they are suitably arranged in a priority order [3]. For example, assume that two secondary tasks, involving all the variables of the system, are assigned in the form: $\sigma_a = f_a(q, \eta)$, $\sigma_b = f_b(q, \eta)$, where $q = [q_1^T \dots q_{n_f}^T]^T$ and $\eta = [\eta_1^T \dots \eta_{n_f}^T]^T$ are the stacked vector of joint and contact variables. Adopting the *augmented projection method* [3], equation (4.10) must be replaced by

$$\begin{bmatrix} \dot{q}_i \\ \dot{\eta}_i \end{bmatrix} = \tilde{J}^\dagger G_i^T (v_o + K e_{o_i}) + N_{o_i} J_a^\dagger K_a e_a + N_{i_{ab}} J_b^\dagger K_b e_b, \quad (4.13)$$

where J_a and J_b are the Jacobian matrices of the secondary tasks, $e_a = \sigma_{a_d} - f_a(q)$ and $e_b = \sigma_{b_d} - f_b(q)$, being σ_{a_d} and σ_{b_d} the desired values of the subtasks, N_a is the projector in the null space of J_a , and $N_{i_{ab}}$ is the projector in the null space of the matrix

$$J_{i_{ab}}(q, \eta) = \begin{bmatrix} \tilde{J}^T(q_i, \eta_i) & J_a^T(q, \eta) & J_b^T(q, \eta) \end{bmatrix}^T.$$

The components of the equation (4.13) corresponding to the actuated joints represent the developed kinematic control law, and they will be the inputs of the low-level controller.

A similar scheme can be adopted for the transpose-based CLIK algorithm in the case of multiple secondary tasks.

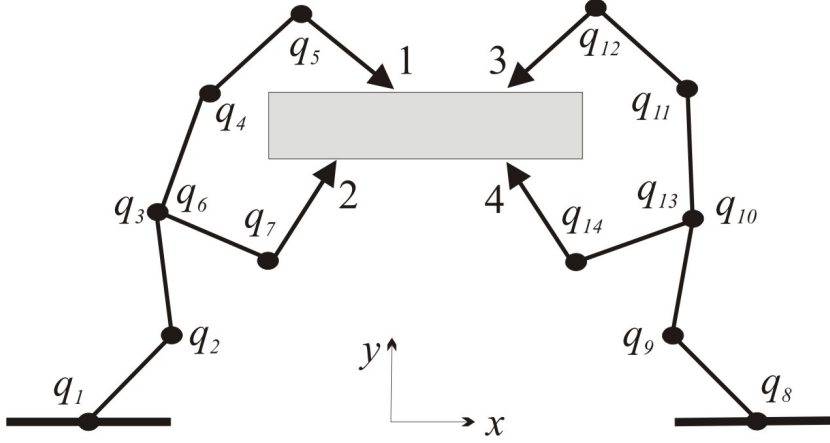


Figure 4.3: Manipulation system for the case study.

4.5 Simulation of the case study

The CLIK scheme with Jacobian pseudo-inverse has been tested on a manipulation system, represented in Figure 4.3, composed by two identical planar grippers, each with two branches and 7 DOFs, grasping a rectilinear bar. For simplicity, all the links have the same length $l = 1$ m. The idea is that of performing a simplified bimanual manipulation task.

It is assumed that, in the initial configuration (reproduced in Figure 4.3), the system grasps the object with tips 3 and 4 aligned to y axis, and with tips 1 and 3, as well as tips 2 and 4, aligned to x axis. The distance between tips 1 and 3 is 0.7 m, while the distance between points 2 and 4 is 1.2 m.

The contact at point 1 and 3 is assumed of type “point contact without friction”, while the contact at points 2 and 4 is of type “hard finger with friction”, i.e., the surface of the object is smooth on the top side and rough on the bottom side. This implies that two contact variables θ_i and ξ_i are required to represent rotation and sliding of finger i ($i = 1, 2$) on the object’s surface, while two contact variables θ_2 and θ_4 have to be introduced to represent the rotation of finger i ($i = 2, 4$) with respect to the surface of the object. It is easy to verify that this grasp is force closure [67] and kinematically determinate [66].

The manipulation system has a total of 20 DOFs that are not all independent, for the presence of 9 closed-chain kinematic constraints; the resulting 11 DOFs can be exploited to satisfy a certain number of tasks.

The main task consists in a desired trajectory for the position of the object, and a desired constant orientation with the bar aligned to x axis. The position path, represented in Fig. 4.4, can be decomposed in the following sequence:

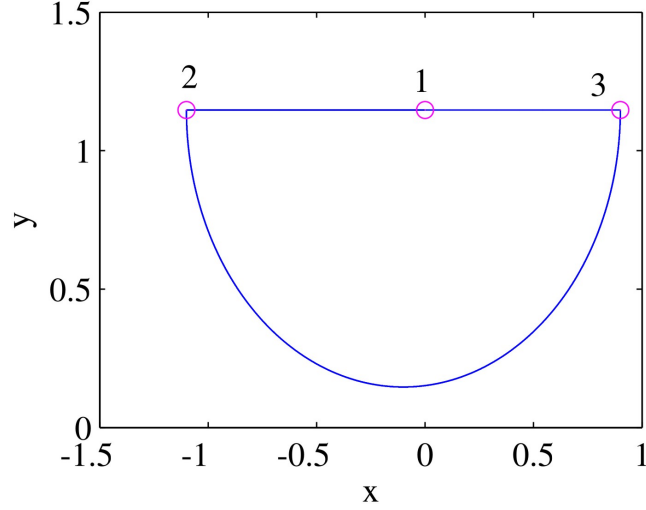


Figure 4.4: Path imposed to the position of the object.

- line segment 1-2;
- arc segment 2-3;
- line segment 3-2;
- arc segment 2-3.

The time law for each segment is a fifth-order polynomial with null first and second derivatives at initial and final time, of a 10 s duration. A 1 s pause is present before the execution of each segment.

Three simple secondary tasks, with decreasing priorities, are considered, according to the augmented projection method. Namely:

- a. *joint limits*: a constraint $q_3 \leq \pi/2$ is imposed to joint variable q_3 ;
- b. *collision avoidance*: the distance between fingers 1 and 3, which can slide on the surface, must be greater than a threshold $d = 0.3$ m;
- c. *grasp quality*: the contact points of fingers 1 and 3 are to be kept as close as possible to the middle point between the contact points of fingers 2 and 4.

Notice that these tasks do not saturate all the available DOFs of the system. Additional tasks could be imposed, but they are not considered here for brevity.

The secondary tasks are not all active at the same time, but they start on the basis of a threshold mechanism. Namely, subtask **a** is active only when q_3 is in

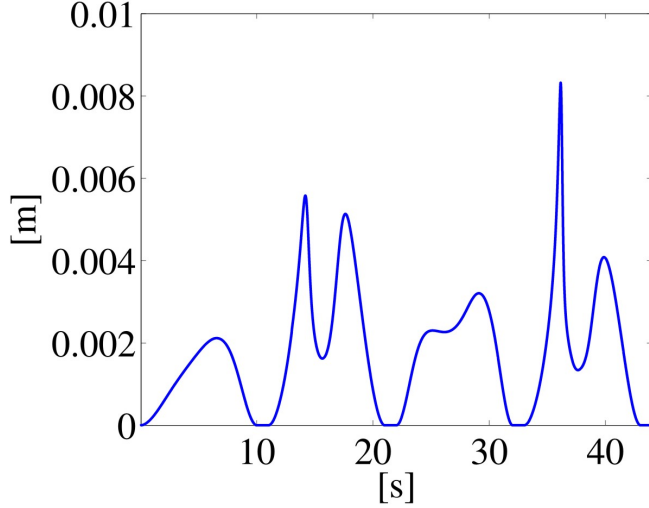


Figure 4.5: Time history of the object's pose error in terms of the average of the norm of the pose errors of the 4 parallel CLIK algorithms. Dashed line: without secondary tasks. Continuous line: with secondary tasks.

the neighbor of $\pi/2$, while subtask **b** is active when the distance between tips 1 and 3 is lower than 0.4 m. Moreover, the gains of the subtasks are not constant; in detail, the gain of subtask **a** depends on the value of q_3 , the gain of subtask **b** depends on the norm of the distance between fingertips 1 and 3, and the gain of subtask **c** depends on the distance of the fingertips 1 and 3 from the center of the tips 2 and 4.

It is straightforward to notice that the augmented Jacobian must change according to the variation of the various subtasks. To ensure smooth transitions between the changes of the subtasks, methods proposed in [63] can be used.

Two different simulations have been made, with and without the presence of secondary tasks. The gain matrix for the main task in (4.13) is chosen as $\mathbf{K} = 100\mathbf{I}$.

The results of Figure 4.5 show the norm of the object's pose error with and without secondary tasks. It can be observed that the error in the two cases is the same, because the secondary tasks are in the null space of the main task.

The time histories of the significant variables for the secondary tasks are reported in Figures 4.6 – 4.7.

Figure 4.6 shows the time history of the joint 3 variable, assuming that, at time $t = 0$, this variable is close to the upper joint limit ($\pi/2$). It can be verified that, without secondary tasks, the joint variable violates the limit, differently from the

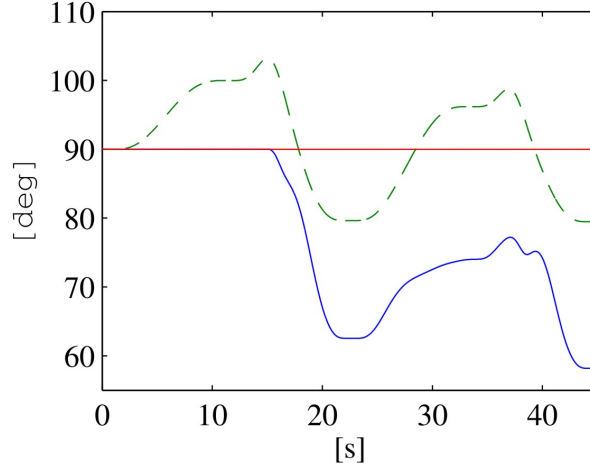


Figure 4.6: Time history of joint 3 variable. Dashed line: without secondary tasks. Continuous line: with secondary tasks.

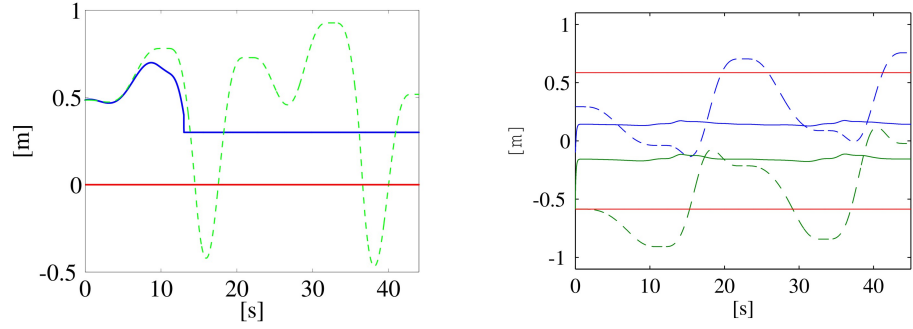


Figure 4.7: On the left: time history of the distance between tips 1 and 3. Dashed line: without secondary tasks. Continuous line: with secondary tasks. On the right: time history of the x position of the contact points with respect to the center of the object. Dashed line: without secondary tasks. Continuous line: with secondary tasks.

case when the joint limit constraint is imposed as secondary task.

Figure 4.7 reports the time history of the distance between tips 1 and 3. It can be seen that, without secondary tasks, the distance between contact points 1 and 3 changes sign, meaning that fingers 1 and 3 overlap. On the other hand, using redundancy, the distance between fingers 1 and 3 remains always positive, as requested by the collision avoidance task between tips 1 and 3. The same figure

reports also the time history of the x position of all the contact points with respect to the center of the object. The constant lines are the position of fixed contact points 2 and 4. It can be seen that, without secondary tasks, the sliding contact points 1 and 3 have large displacements on the surface of the object. On the other hand, using redundancy, they remain close to each other as imposed by the secondary task **c**, without violating task **b**.

Chapter 5

Conclusion and future research directions

A brief recap about the methods presented in this thesis and the achieved results will be the object of the current chapter. Proposals for future research directions will be discussed as well.

5.1 Main results

A task about grasping and manipulation of objects with a multi-fingered robotic hand in unknown environments has been carried out during this thesis. The problem has been obviously split in three different parts, which are namely:

- Detection and reconstruction of the surface of the unknown object.
- Planning of the grasp;
- Manipulation of the object in a coordinate and dexterous way, exploiting the redundancy of the system “robotic multi-fingered hand plus object”.

For each of the previous items, a new method has been proposed and validated with several experiments (for the first two parts) and simulations.

In particular, a new method to control the motion of a multi-fingered hand to achieve a human-like grasp of unknown objects has been presented, which is composed of an iterative object’s surface reconstruction algorithm and of a local optimal grasp planner – combined with a kinematic motion controller –, evolving in a synchronized parallel way. The former algorithm uses an elastic ellipsoidal reconstruction surface, whose axes and dimensions are assigned by a new proposed preshaping process, that is let to evolve dynamically under the action of reconstruction forces. The reconstruction surface shrinks towards the object until some parts of the surface intercept the visual hull of the object. The latter algorithm

moves the fingertips of the robotic multi-fingered hand on the current available reconstruction surface towards points which are optimal (in a local sense) with respect to a certain number of indices weighting both the quality of the grasp and the kinematic configuration of the robotic hand. It is worth noticing that, in this work, a new heuristic grasp quality index has been proposed, but any other quality measures may be chosen in substitution of the proposed one, without effecting the general framework: the only thing to point out is that any other index should just ensure that it can be applied to discretized surfaces of the objects, as shown in the Appendix A. At the end, a control module must ensure that the references given by the planner are correctly followed by the robotic hand.

Once the unknown object has been grasped, one could assign to it a desired motion for a particular task. In such a case, a framework for a coordinate and dexterous manipulation should be considered. For this reason, a kinematic model for motion coordination of a redundant multi-fingered robotic hand has been derived, which allows to compute the object's pose from the joint variables of each finger of the robotic hand, as well as from a suitable set of contact variables. Moreover, a prioritized inverse kinematics scheme with redundancy resolution, both with inverse and transpose Jacobian matrix, has been developed. This algorithm can be used for kinematic control as well as for a local planning method for dexterous manipulation.

5.2 Proposals for the future

Likewise performed in the previous section of this chapter, the directions for future researches can be also split in three parts.

For what concerning the reconstruction of objects' surfaces, proposals are leading at overcoming some assumptions given in the section 3.1. In particular, to satisfy some possible industrial requirements, the object could be also in movement with respect to the base frame and some research directions are trying to overcome the problem about the topology of the object. Some results about this last aspect have been already reached, but this goes beyond the goals of this thesis.

About the quality measures for the grasp, several efforts are addressed in the fulfilment of global optimal grasps instead of just local optima. To accomplish this task, some studies aimed at employing touch sensors to refine the final grasp have been carried out.

For what concerning the kinematic control method for dexterous manipulation, the case of study of Figure 4.3 has been built by some students at Università of Basilicata under the supervision of Prof. Caccavale and Dr. Pierri. A collaboration has been established with this research team and some experimental works will be tested under this new experimental platform. In particular, the contact between the fingertips of the robotic hand and the object will be now dynamic through

the use of soft-pads; further, friction will be inserted in the proposed framework to deal with sliding contact issues. Moreover, other subtasks like those proposed in [76] will be exploited by the redundancy management of the system.

Appendix A

Appendix

As said at the beginning of Chapter 3, many other quality measures may be chosen in substitution of the proposed one in Chapter 3, without effecting the general proposed framework. The only thing to point out is that any other index should just ensure that it can be applied to discretized surfaces of the objects.

In this appendix, this last statement will be proven by showing another grasp quality measure. The method proposed in [64] is hence adopted, suitably modified to cope with the discretization of the grasp configurations, assuming neglectable moments and transversal forces. A three-fingered robotic hand will be considered throughout this Appendix.

A.1 Mirtich and Canny grasp quality index

Let $\mathbf{w} = [\mathbf{f}^T \quad \boldsymbol{\mu}^T]^T$ denote the wrench vector collecting the force \mathbf{f} and moment $\boldsymbol{\mu}$. Assuming that the finger forces are applied along the directions normal to the object's surface, the direction of the force is then specified only by the contact point.

Let \mathcal{W} denote the space of wrenches, $\mathcal{W}_f \subset \mathcal{W}$ the space of unit forces acting in the grip plane, which is the plane containing the three contact points, through the center of grip, $\mathcal{W}_{\perp\mu} \subset \mathcal{W}$ the space of pure moments acting along the direction perpendicular to the grip plane. Moreover, let $\mathbf{g}^{-1}(-\mathbf{w})$ denote the set of finger forces which can resist the external wrench \mathbf{w} .

Finally, consider the quantity

$$Q_1 = \min_{\mathbf{w} \in \mathcal{W}_f} \left(\max_{\mathbf{f} \in \mathbf{g}^{-1}(-\mathbf{w})} \frac{1}{\|\mathbf{f}\|_f} \right),$$

which is a measure of the grasp ability to resist unit forces in the grip plane, and the quantity

$$Q_2 = \min_{\mathbf{w} \in \mathcal{W}_{\perp\mu}} \left(\max_{\mathbf{f} \in \mathbf{g}^{-1}(-\mathbf{w})} \frac{1}{\|\mathbf{f}\|_f} \right),$$

which is a measure of the grasp ability to resist unit moments normal to the grip plane.

The optimal grasp proposed in [64] is defined as the grasp that maximizes Q_2 among all grasps which maximize Q_1 .

It can be proven (see [64]) that the optimum grasp with three fingers in a 2-D case under the above optimal criterion is reached when the normal forces are symmetric, with directions spaced 120° apart. Moreover, this grasp maximizes also the size of the outer triangle, defined as the triangle formed by the three lines perpendicular to the normal finger forces passing through the respective contact points. Under the same criterion, the optimum grasp with three fingers in a 3-D case is achieved when the maximum circumscribing prism-shaped grasp, that has the largest outer triangle, is selected among the grasps where the normal finger forces lie within the same grip plane and are in an equilateral configuration.

Therefore, to reach the optimum in the 3-D case with three fingers, the planner has to seek three points in equilateral configuration on the object's surface, so that the normal forces lie in the same grip plane, and for which the circumscribing prism grasp is maximum.

A.2 Re-arrangement of the Mirtich and Canny quality index

Since the reconstructed surface of the object is sampled by points/masses, the above method cannot be directly applied. Differently from the continuous case, due to the presence of a finite set of sampled points, the existence of a “grip plane” containing all the normal forces is not guaranteed. This is mainly due to the fact that, because of the discretization, the normals to the surface are an approximation of the real ones. Considering that the optimal criterion requires that the desired normals have to be spaced 120° apart, a discretized implementation of the method of [64] is hence here proposed.

For each candidate configuration of three grasp points, the normal directions are estimated on the basis of the available point-wise approximation of the surface. Then, the unit vector normal to the grip plane containing the three points is evaluated. Denoting with ϑ_j the angle between the direction of the normal force applied to point j and the direction normal to the grip plane, a Coplanarity Error Index (CEI) can be defined as follow:

$$CEI = \frac{\sum_{j=1}^3 |\vartheta_j - 90^\circ|}{3}.$$

Obviously, the closer CEI to zero, the more the normal forces lie in the same plane. The definition of a threshold Φ_{CEI} allows discarding all those configurations having

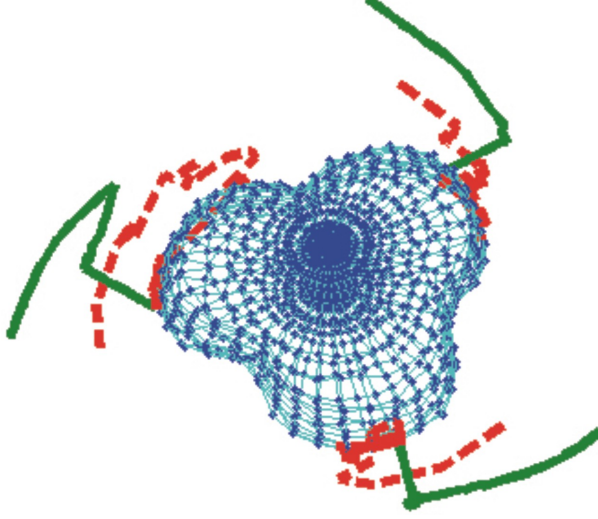


Figure A.1: Finger trajectories evaluated by the local grasp planner (continuous lines) and the corresponding sequence of grasp points on the reconstructed object's surface (dotted lines).

a value of CEI higher than Φ_{CEI} ; hence, all the remaining grasp configurations are assumed to have forces lying in the same grip plane and can be further processed.

The next step consists in looking for an equilateral grasp configuration. To this aim, for each grasp configuration, the unit vector normal to the object's surface at each contact point is projected on the grip plane. Denoting with φ_j the angle between these projections for each of the 3 couple of points of the considered configuration, an Equilateral Error Grasp Index ($EEGI$) can be defined as:

$$EEGI = \frac{\sum_{j=1}^3 |\varphi_j - 120^\circ|}{3}.$$

Clearly, the closer $EEGI$ to zero, the nearer the configuration to an equilateral grasp. The definition of a threshold Φ_{EEGI} allows discarding all those configurations with a value of $EEGI$ higher than Φ_{EEGI} ; hence, all the remaining grasp configurations are assumed to be equilateral.

Among all the equilateral configurations, the maximum circumscribing prism has to be found; if the grasp configuration associated with the largest prism is different from the current target configuration, this is taken as the new grasp configuration.

Notice that, in the case that the grasp configuration changes, the whole process starts again with the new contact points, by considering the new contours and applying the complete sequence of index-based tests. The algorithm stops if the

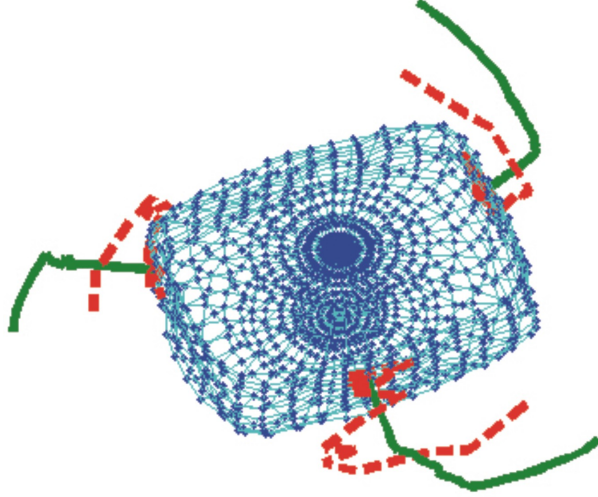


Figure A.2: Finger trajectories evaluated by the local grasp planner (continuous lines) and the corresponding sequence of grasp points on the reconstructed surface for the smooth prism (dotted lines).

best grasp configuration remains unchanged at the end of the optimization, or in the case that all the candidate grasp configurations are discarded during the process. Moreover, it is worth noticing that also some other testes based on the kinematics of the hand in the current configuration can be carried out for the research of the (local) optimum grasp under this proposed grasp quality measure: for more details see [57].

A.3 Simulations

The proposed quality index has been tested with simulations using synthesized objects.

The dynamic parameters of the reconstruction ellipsoid have been chosen as follows: $m = 10^{-3} \text{ kg}$, $k = 0.3 \cdot 10^{-3} \text{ N/m}$, $b = 0.09 \cdot 10^{-3} \text{ Ns/m}$, and $F_a = 5 \text{ N}$.

By setting $\Phi_{CEI} = 15^\circ$ and $\Phi_{EEGI} = 10^\circ$, the configuration of the grasp and the trajectories of the fingers for the first object are those of Fig. A.1. Moreover, the final configuration of the grasp (which can be proven to be the global optimal grasp configuration) is characterized by the values $CEI = 13^\circ$ and $EEGI = 2.2^\circ$.

A prism with smooth lateral corners has also been considered. The configuration of the grasp and the corresponding trajectories are shown in Fig. A.2. The values $CEI = 9.9^\circ$ and $EEGI = 1.04^\circ$ are obtained in the final configuration. Remarkably, an equilateral symmetry is achieved for the final grasp: two fingers

are placed on the smooth corners, and the other finger is placed in the middle of the opposite surface. This configuration corresponds to an opposite grasp ensuring force closure; as before, it can be proven that this grasp is optimal also in a global sense, although the proposed approach can only guarantee that a local optimum is achieved.

Bibliography

- [1] S.P. Ananthanarayanan and A.A. Gldenberg, *Dexterous manipulation using qualitative reasoning, part 1: determination of a desired object displacement*, Fifth International Conference on Advanced Robotics (Pisa), 1991.
- [2] ———, *Dexterous manipulation using qualitative reasoning, part 1: modelling and synthesis of finger manipulative synergies*, Fifth International Conference on Advanced Robotics (Pisa), 1991.
- [3] G. Antonelli, *Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotics systems*, IEEE Transactions on Robotics **25** (2009), no. 5, 985–994.
- [4] K.K. Aydin, *Fuzzy logic, grasp preshaping for robot hands*, IEEE Annual Conference of the North America Fuzzy Information Processing Society (College Park), 1995.
- [5] A. Bemporad and M. Morari, *Control of systems integrating logic, dynamics, and constraints*, Automatica **35** (1999), no. 3, 407–427.
- [6] H. Berghius, R. Ortega, and H. Nijmeijer, *A robust adaptive controller for robot manipulators*, IEEE International Conference on Robotics and Automation (Nice), 1992.
- [7] L. Biagiotti, H. Liu, G. Hirzinger, and C. Melchiorri, *Cartesian impedance control for dexterous manipulation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas), 2003.
- [8] A. Bicchi and D. Prattichizo, *Manipulability of cooperative robots with unactuated joints and closed-chain mechanisms*, IEEE Transactions on Robotics and Automation **16** (2000), no. 4, 336–345.
- [9] I. Biederman, *Recognition-by-components: a theory of human image understanding*, Psychological Review (1987).
- [10] F. Bley, V. Schmirgel, and K.F. Kraiss, *Mobile manipulation based on generic object-knowledge*, IEEE International Symposium on Robot and Human Interactive Communication (Hatfield), 2006.

- [11] C. Borst, M. Fischer, and G. Hirzinger, *Calculating hand configurations for precision and pinch grasps*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Lausanne), 2002.
- [12] M. Buss, H. Hashimoto, and J.B. Moore, *Dexterous hand grasping force optimization*, IEEE Transactions on Robotics and Automation **12** (1996), no. 3, 406–418.
- [13] A. Chella, H. Dindo, F. Matraxia, and R. Pirrone, *A neuro-genetic approach to real-time visual grasp synthesis*, International Joint Conference on Neural Networks (Orlando), 2007.
- [14] J. Chen and M. Zribi, *Control of multifingered robot hands with rolling and sliding contacts*, The International Journal of Advanced Manufacturing Technology **16** (2000).
- [15] E. Chinellato, R.B. Fisher, A. Morales, and P. del Pobil, *Ranking planar grasp configurations for a three-fingered hand*, IEEE International Conference on Robotics and Automation (Taipei), 2003.
- [16] H.R. Choi, W.K. Chung, and Y. Youm, *Stiffness analysis and control of multi-fingered robot hands*, Journal of Dynamic Systems, Measurement, and Control **117** (1995).
- [17] W.Y. Chung and K.J. Waldron, *Simulation of dexterous manipulation for multifingered systems*, IEEE International Conference on Robotics and Automation (San Diego), 1994.
- [18] R. Cipolla and A. Blake, *Surface shape from the deformation of apparent contours*, International Journal of Computer Vision **9** (1992), no. 2, 83–112.
- [19] L.D. Cohen, *On active contour models and balloons*, Computer Vision, Graphics, and Image Processing: Image Understanding **53** (2001), no. 2, 512–522.
- [20] A.A. Cole, P. Hsu, and S.S. Sastry, *Dynamic control of sliding by robot hands for regrasping*, IEEE Transactions on Robotics and Automation **8** (1992), no. 1, 42–52.
- [21] A.B.A. Cole, J.E. Hauser, and S.S. Sastry, *Kinematics and control of multifingered hands with rolling contact*, IEEE Transactions on Automatic Control **34** (1989), no. 4, 398–407.
- [22] M.R. Cutkosky, *On grasp choice, grasp models, and the design of hands for manufacturing tasks*, IEEE Transactions on Robotics and Automation **5** (1989), no. 3, 269–279.

- [23] M.R. Cutkosky and J.M. Hyde, *Manipulation control with dynamic tactile sensing*, 6th International Symposium of Robotic Research, 1993.
- [24] M.R. Cutkosky and I. Kao, *Computing and controlling the compliance of a robotic hand*, IEEE Transactions on Robotics and Automation **5** (1989), no. 2, 151–165.
- [25] A. DasGupta and H. Hatwal, *Dynamics and nonlinear coordination control of multifingered mechanical hands*, Journal of Dynamic Systems, Measurement, and Control **120** (1998), no. 2, 275–281.
- [26] D. Ding, Y.H. Liu, and S. Wang, *Computation of 3-d form-closure grasps*, IEEE Transactions on Robotics and Automation **17** (2001), no. 4, 512–522.
- [27] T. Dorsam, S. Fatikow, and I. Streit, *Fuzzy-based grasp-force-adaptation for multifingered robot hands*, IEEE Conference on Fuzzy Systems (Orlando), 1993.
- [28] Z. Doulgeri, J. Fasoulas, and S. Arimoto, *Feedback control for object manipulation by a pair of soft tip fingers*, Robotica **20** (2002), no. 1, 1–11.
- [29] C. Dune, E. Marchand, C. Collwet, and C. Leroux, *Active rough shape estimation of unknown objects*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Nice), 2008.
- [30] C.R. Dyer, *Volumetric scene reconstruction from multiple views*, Foundations of Image Analysis (L.S. Davis, ed.), Kluwer.
- [31] J. Fasoulas and Z. Doulgeri, *Object stable grasping control by dual robotic fingers with soft rolling contacts*, IEEE International Conference on Robotics and Automation (Washington), 2002.
- [32] C. Ferrari and J. Canny, *Planning optimal grasps*, IEEE International Conference on Robotics and Automation (Nice), 1992.
- [33] M. Fischer and G. Hirzinger, *Fast planning of precision grasps for 3d objects*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Grenoble), 1997.
- [34] F. Fleischer, A. Casile, and M.A. Giese, *View-independent recognition of grasping actions with a cortex-inspired model*, IEEE International Conference on Humanoid Robots (Paris), 2009.
- [35] J.S. Franco and E. Boyer, *Exact polyhedral visual hulls*, British Machine Vision Conference, 2003.

- [36] Y. Guan and H. Zhang, *Kinematic feasibility analysis of 3d grasps*, IEEE International Conference on Robotics and Automation (Seoul), 2001.
- [37] L. Han, Z. Li, J.C. Trinkle, Z. Qin, and S. Jiang, *The planning and control of robot dexterous manipulation*, IEEE International Conference on Robotics and Automation (San Francisco), 2000.
- [38] L. Han and J.C. Trinkle, *The instantaneous kinematics of manipulation*, IEEE International Conference on Robotics and Automation (Leuven), 1998.
- [39] R.D. Hester, M. Cetin, C. Kapoor, and D. Tesar, *A criteria-based approach to grasp synthesis*, IEEE International Conference on Robotics and Automation (Detroit), 1999.
- [40] R.A. Hilhorst and K. Tanie, *Dexterous manipulation of objects with unknown parameters by robot hands*, IEEE International Conference on Robotics and Automation (San Diego), 1994.
- [41] G. Hinton, *Some computational solution to berenstein's problems*, Human Motor Actions Bernstein Reassessed (H.T.A. Whiting, ed.), Elsevier Science Publishers.
- [42] T.H. Hong and M.O. Shneier, *Describing a robot's workspace using a sequence of views from a moving camera*, IEEE Transactions on Pattern Analysis and Machine Intelligence **7** (1985), no. 6, 721–726.
- [43] R.D. Howe and M.R. Cutkosky, *Integrating tactile sensing with control for dexterous manipulation*, IEEE International Workshop on Intelligent Motion Control (Istanbul), 1990.
- [44] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky, *Grasping, manipulation, and control with tactile sensing*, IEEE International Conference on Robotics and Automation (Cincinnati), 1990.
- [45] Y. Hu, R. Eagleson, and M.A. Goodale, *Human visual servoing for reaching and grasping: the role of 3-d geometric features*, IEEE International Conference on Robotics and Automation (Detroit), 1999.
- [46] M. Huber and R.A. Grupen, *Robust finger gaits from closed-loop controllers*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Lausanne), 2002.
- [47] K. Huebner and D. Kragic, *Selection of robot pre-grasps using box-based shape approximation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Nice), 2008.

-
- [48] J.M. Hyde, M.R. Tremblay, and M.R. Cutkosky, *An object-oriented framework for event-driven dextrous manipulation*, 4th International Symposium of Experimental Robotics, 1995.
- [49] K. Imazeki and T. Maeno, *Hierarchical control method for manipulation/grasping tasks using multi-fingered robot hand*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas), 2003.
- [50] A. Isidori, *Nonlinear control systems*, Springer-Verlag, London, 1989.
- [51] O. Khatib and J. Burdick, *Motion and force control of robot manipulators*, IEEE International Conference on Robotics and Automation, 1986.
- [52] B.H. Kim, B.J. Yi, S.R. Oh, and H.H. Suh, *Independent finger and independent joint-based compliance control of multifingered robot hands*, IEEE Transactions on Robotics and Automation **19** (2003), no. 2, 185–199.
- [53] Y. Kim and M.A. Minor, *Distributed kinematic motion control of multi-robot coordination subject to physical constraints*, The International Journal of Robotic Research **29** (2010), no. 1, 92–109.
- [54] K.N. Kutulakos and S.M. Seitz, *A theory of shape by space carving*, International Journal of Computer Vision **38** (2000), no. 3, 199–218.
- [55] Z. Li and S.S. Sastry, *Task-oriented optimal grasping by multi-fingered robot hands*, IEEE Journal of Robotics and Automation **4** (1988), no. 1, 32–44.
- [56] V. Lippiello and F. Ruggiero, *Surface model reconstruction of 3d objects from multiple views*, IEEE International Conference on Robotics and Automation (Kobe), 2009.
- [57] V. Lippiello, F. Ruggiero, and B. Siciliano, *Floating visual grasp of unknown objects using an elastic reconstruction surface*, International Symposium of Robotic Research (Lucerne), 2009.
- [58] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, *Human-like visual grasp of unknown objects*, International Conference on Applied Bionics and Biomechanics (Venice), 2010.
- [59] ———, *Preshaped visual grasp of unknown objects with a multi-fingered hand*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Taipei), 2010.
- [60] V. Lippiello, F. Ruggiero, and L. Villani, *Exploiting redundancy in closed-loop inverse kinematics for dexterous object manipulation*, IEEE International Conference on Advanced Robotics (Munich), 2009.

- [61] ———, *Floating visual grasp of unknown objects*, IEEE/RSJ International Conference on Intelligent Robots and Systems (St. Louis), 2009.
- [62] ———, *Inverse kinematics for object manipulation with redundant multi-fingered robotic hands*, International Workshop on Robot Motion and Control (Czerniejewo), 2009.
- [63] N. Mansard and F. Chaumette, *Task sequencing for high-level sensor-based control*, IEEE Transactions of Robotics and Automation **23** (2007), no. 1, 60–72.
- [64] B. Mirtich and J. Canny, *Easily computable optimum grasps in 2-d and 3-d*, IEEE International Conference on Robotics and Automation (San Diego), 1994.
- [65] D. Montana, *The kinematics of contact and grasp*, The International Journal of Robotic Research **7** (1988), no. 3, 17–32.
- [66] ———, *The kinematics of multi-fingered manipulation*, IEEE Transactions on Robotics and Automation **11** (1995), no. 4, 491–503.
- [67] R.M. Murray, Z.X. Li, and S.S. Sastry, *A mathematical introduction to robotic manipulation*, CRC press, Boca Raton, 1993.
- [68] K. Nagai and T. Yoshikawa, *Dynamic manipulation/grasping control of multi-fingered robot hands*, IEEE International Conference on Robotics and Automation (Atlanta), 1993.
- [69] ———, *Grasping and manipulation by arm/multifingered-hand mechanism*, IEEE International Conference on Robotics and Automation (Nagoya), 1995.
- [70] T.N. Nguyen and H.E. Stephanou, *A continuous model of robot hand pre-shaping*, IEEE International Conference on Systems, Man and Cybernetics (Cambridge), 1989.
- [71] H. Noborio, S. Fukuda, and S. Arimoto, *Construction of the octree approximating three-dimensional objects by using multiple views*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988), no. 6, 769–782.
- [72] A.M. Okamura, N. Smaby, and M.R. Cutkosky, *An overview of dextrous manipulation*, IEEE International Conference on Robotics and Automation (San Francisco), 2000.
- [73] B. Ozyer and E. Oztop, *Task dependent human-like grasping*, IEEE International Conference on Humanoid Robots (Daejeon), 2008.

-
- [74] D. Perrin, C.E. Smith, O. Masoud, and N.P. Papanikolopoulos, *Unknown object grasping using statistical pressure models*, IEEE International Conference on Robotics and Automation (San Francisco), 2000.
- [75] R. Platt, A.H. Fagg, and R.A. Grupen, *Manipulation gaits: sequences of grasp control tasks*, IEEE International Conference on Robotics and Automation (New Orleans), 2004.
- [76] ———, *Null-space grasp control: theory and experiments*, IEEE Transactions on Robotics **26** (2010), no. 2, 282–295.
- [77] R.J. Platt, *Learning and generalizing control-based grasping and manipulations skills*, Ph.D. thesis, University of Massachusetts Amherst, 2006.
- [78] N.S. Pollard, *Synthesizing grasps from generalized prototypes*, IEEE International Conference on Robotics and Automation (Minneapolis), 1996.
- [79] J. Ponce and B. Faverjon, *On computing three-finger force-closure grasps of polygonal objects*, IEEE Transactions on Robotics and Automation **11** (1995), no. 6, 868–881.
- [80] S. Pragoonwit and R. Benjamin, *3d surface point and wireframe reconstruction from multiview photographic images*, Image and Vision Computing **25** (2007).
- [81] M. Prats, P.J. Sanz, and A.P. del Pobil, *Task-oriented grasping using hand preshapes and tasks frames*, IEEE International Conference on Robotics and Automation (Rome), 2007.
- [82] D. Prattichizzo and J.C. Trinkle, *Grasping*, Handbook of Robotics (B. Siciliano and O. Khatib, eds.), Springer-Verlag, 2008, pp. 671–700.
- [83] C. Remond, V. Perdureau, and M. Drouin, *A hierarchical multi-fingered hand control structure with rolling contact compensation*, IEEE International Conference on Robotics and Automation (Washington), 2002.
- [84] M.A. Roa and R. Suarez, *Geometrical approach for grasp synthesis on discretized 3d objects*, IEEE/RSJ International Conference on Intelligent Robots and Systems (San Diego), 2007.
- [85] T. Schlegl, M. Buss, T. Omata, and G. Schmidt, *Fast dextrous regrasping with optimal contact forces and contact sensor-based impedance control*, IEEE International Conference on Robotics and Automation (Seoul), 2001.
- [86] S.A. Schneider and R.H. Cannon Jr., *Object impedance control for cooperative manipulation: theory and experimental results*, IEEE Transactions on Robotics and Automation **8** (1992), no. 3, 383–394.

- [87] S.M. Seitz and C.R. Dyer, *Photorealistic scene reconstruction by voxel coloring*, International Journal of Computer Vision **35** (1999), no. 2, 151–173.
- [88] K.B. Shimoga, *Robot grasp synthesis algorithms: a survey*, The International Journal of Robotic Research **15** (1996), no. 3, 230–266.
- [89] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics. modelling, planning and control*, Springer, London, 2008.
- [90] J.J. Slotine and W. Li, *On adaptive control of robot manipulators*, The International Journal of Robotic Research **6** (1987), no. 3, 49–59.
- [91] J.S. Son and R.D. Howe, *Performance limits and stiffness control of multi-fingered hands*, International Symposium on Experimental Robotics, 1995.
- [92] T.H. Speeter, *Modal state position controller for the utah/mit dextrous hand*, IEEE Transactions on Robotics and Automation **6** (1990), no. 2, 274–277.
- [93] S. Stramigioli, *Modeling and ipc control of interactive mechanical systems*, Springer-Verlag, London, 2001.
- [94] S. Stramigioli, C. Melchiorri, and S. Andreotti, *A passivity-based control scheme for robotic grasping and manipulation*, IEEE International Conference on Decision and Control (Phoenix), 1999.
- [95] R. Suarez, M. Roa, and J. Cornella, *Grasp quality measures*, Technical report ioc-dt-p-2006-10, Universitat Politècnica de Catalunya, Institut d’Organització i Control de Sistemes Industrials, 2006.
- [96] T. Supuk, T. Kodek, and T. Bajd, *Estimation of hand preshaping during human grasping*, Medical Engineering and Physics **27** (2005).
- [97] R. Szeliski, *Rapid octree construction from image sequences*, CVGIP: Image Understanding **58** (1993), no. 1, 23–32.
- [98] T. Takahashi, T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, M. Iribe, T. Fukushima, and M. Fujita, *Adaptive grasping by multifingered hand with tactile sensor based on robust force and position control*, IEEE International Conference on Robotics and Automation (Pasadena), 2008.
- [99] K. Terashima, T. Kondo, P. Minyong, T. Miyoshi, and H. Kitagawa, *Sense feedback control of human muscle by multi-fingered robot hand*, 16th International Federation of Automatic Control Congress, 2005.
- [100] S. Ueki, H. Kawasaki, and T. Mouri, *Adaptive coordinated control of multi-fingered hands with sliding contacts*, SICE-ICASE International Joint Conference (Busan), 2006.

-
- [101] T. Wimboeck, C. Ott, and G. Hirzinger, *Passivity-based object-level impedance control for a multifingered hand*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Beijing), 2006.
- [102] ———, *Analysis and experimental evaluation of the intrinsically passive controller (ipc) for multifingered hands*, IEEE International Conference on Robotics and Automation (Pasadena), 2008.
- [103] S.A. Winges, D.J. Weber, and M. Santello, *The role of vision on hand pre-shaping during reach to grasp*, Experimental Brain Research **152** (2003), no. 4, 489–498.
- [104] C. Xu and J.L. Prince, *Snakes, shapes, and gradient vector flow*, IEEE Transactions on Image Processing **7** (1998), no. 3, 359–369.
- [105] A. Yezzi, G. Slabaugh, Broadhurst A, R. Cipolla, and R. Schafer, *A surface evolution approach to probabilistic space carving*, International Symposium on 3D Data Processing, Visualization, and Transmission, 2002.
- [106] Y. Yin and S. Hosoe, *Tracking control of discrete and continuous hybrid systems: modeling and servoing problem of dexterous hand manipulation*, IEEE International Conference on Control Applications (Taipei), 2004.
- [107] ———, *Mld modeling and optimal control of hand manipulation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Edmonton), 2005.
- [108] Y. Yin, Z. Luo, M. Svinin, and S. Hosoe, *Hybrid control of multi-fingered robot hand for dexterous manipulation*, International Conference on Systems, Man and Cybernetics, 2003.
- [109] Y. Yin and M. Svinin, *Modeling and control of multifingered robot hand for dexterous manipulation: a continuous and discrete hybrid approach*, International Conference on Systems, Man and Cybernetics, 2004.
- [110] T. Yoshikawa, M. Koeda, and H. Fujimoto, *Shape recognition and grasping by robotic hands with soft fingers and omnidirectional camera*, IEEE International Conference on Robotics and Automation (Pasadena), 2008.
- [111] B.H. Yoshimi and P.K. Allen, *Visual control of grasping and manipulation tasks*, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (Las Vegas), 1994.
- [112] R. Zollner, O. Rogalla, R. Dillmann, and M. Zollner, *Understanding users intention: programming fine manipulation tasks by demonstration*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Lausanne), 2002.